Universidad de Valladolid
Facultad de Ciencias

Trabajo Fin de Grado

Grado en Matemáticas

Métodos numéricos para problemas de mínimos cuadrados y ajuste de parámetros en modelos matemáticos de memristores

Autora: Cristina Santa Cruz González

Tutora: María Paz Calvo Cabrero

"Me gustaría agredecer la gran dedicación con la que mi tutora Mari Paz Calvo me ha guiado en la realización de este Trabajo de Fin de Grado. Las dificultades imprevistas derivadas del confinamiento por el COVID-19 no han sido impedimento para que usando su experiencia y conocimiento me haya ido resolviendo todas las dudas que me iban surgiendo así como orientándome en la manera de proceder en la parte experimental".

"También, quiero agradecer a mis padres y hermanos por su constante apoyo, sobre todo en los momento díficiles, durante todo mi período de formación en la Universidad que se culmina con la presentación de este TFG".

Índice general

Introducción

El problema de mínimos cuadrados es aplicable a múltiples áreas del conocimiento y es el principal recurso para tratar problemas de optimización. Tanto matemáticos, físicos, químicos o economistas usan este tipo de procedimientos al modelizar y así miden las discrepancias entre lo esperado según un modelo teórico y el comportamiento experimental observado. Por medio de la minimización de la denominada función objetivo se pueden determinar los parámetros desconocidos que intervienen en un modelo de tal forma que proporcione un resultado lo más próximo posible al dato real. En este TFG estudiamos y analizamos la robustez de distintos métodos de optimización para tratar especialmente el caso en que los parámetros que intervienen en el modelo teórico con el que se quiere ajustar lo hagan de forma no lineal.

Para el mejor entendimiento de este trabajo es recomendable tener conocimientos previos adquiridos en las asignaturas de "Análisis Numérico" y "Ampliación de Análisis Numérico", aunque también aplicaremos razonamientos de "Análisis Matemático", "Álgebra y Geometría" y "Teoría de la Probabilidad y Estadísitica Matemática".

El trabajo se estructura en tres capítulos. El primero se basa en una profundización del, ya mencionado en varias ocasiones durante la carrera, problema lineal de mínimos cuadrados. Inicialmente, se explica el origen del problema de mínimos cuadrados, haciendo especial hincapié en el caso lineal para el que se expondrán distintos procedimentos para poder resolverlo.

El segundo capítulo se centra en el caso no lineal donde, siguiendo fundamentalmente los libros de Björck [1] y Nocedal [11], se explica las estrategias generales con las que se resuelven estos problemas además de los métodos más utilizados, detallando las ventajas e inconvenientes que poseen según el problema a analizar.

Finalmente, en el tercer capítulo se expone una aplicación real, en el campo de la electrónica, de uno de los métodos explicados en el segundo capítulo, ajustando el modelo matemático desarrollado por E. Miranda en [9] con el que se trata de describir el funcionamiento de un nuevo dispositivo denominado memristor.

De forma complementaria, se incluyen dos apéndices.

El Apéndice A en el que se incluye una serie de propiedades y teoremas, referidos a la pseudoinversa de una matriz, necesarios para poder entender algunos de los razonamientos expuestos en los Capítulos 1 y 2.

El Apéndice B en el que se pueden encontrar los códigos de Matlab utilizados para la realización de las distintas simulaciones descritas en el Capítulo 3.

En Valladolid, a 29 de junio de 2020.

Capítulo 1

Preliminares: El problema de mínimos cuadrados

En este primer capítulo se introduce la estructura y el punto de vista estadístico del problema de mínimos cuadrados y se explica el modo de proceder en el caso en que este problema sea lineal.

1.1. Estructura del problema

El problema de mínimos cuadrados es un problema computacional de vital importancia a la hora de querer ajustar un conjunto de datos, obtenidos al realizar un experimento, a un modelo matemático. Con el objetivo de reducir los posibles errores en la recopilación de estos datos es conveniente recoger un gran número de medidas, frecuentemente consiguiendo un número bastante mayor que los parámetros desconocidos que presenta el modelo matemático al que se va a ajustar. En primer lugar, analizaremos la estructura general del problema de mínimos cuadrados. Para ello, introducimos inicialmente distintos conceptos.

Definición 1.1. Dados unos valores experimentales (y_i, t_i), $i = 1, \ldots, m$, y una función $g(x, t)$ que representa el modelo teórico cuyos parámetros x se desean ajustar, se define la **función objetivo** del problema de mínimos cuadrados como

$$f(x) = \frac{1}{2} \sum_{i=1}^{m} r_i^2(x), \tag{1.1}$$

donde

$$r_i(x) = y_i - g(x, t_i), \quad i = 1, \ldots, m. \tag{1.2}$$

Debemos señalar que m corresponde al número de datos experimentales recopilados y n al número de parámetros que intervienen en el modelo, siendo estos las componentes del vector x.
En los problemas que analizaremos, siempre se considerará que $m \geq n$.

Definición 1.2. Se llama **vector residual** a la aplicación $r : \mathbb{R}^n \to \mathbb{R}^m$ tal que

$$r(x) = (r_1(x), r_2(x), \ldots, r_m(x))^T, \tag{1.3}$$

cuya componente i-ésima representa el error entre la predicción del modelo y el i-ésimo dato experimental observado.
La matriz jacobiana de este vector residual es $J(x) \in \mathbb{R}^{m \times n}$

$$J(x) = \left[\frac{\partial r_i}{\partial x_j}\right]_{\substack{i=1,2,\ldots,m \\ j=1,2,\ldots,n}} = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}, \tag{1.4}$$

donde cada $\nabla r_i(x)$ es el gradiente de r_i. Dicho de otra forma, cada elemento de la matriz jacobiana es $J(x)_{i,j} = \frac{\partial r_i}{\partial x_j}$.

Por otra parte, la matriz hessiana de cada residuo $r_i(x)$ es

$$G_i(x) = \nabla^2 r_i(x) \in \mathbb{R}^{nxn}, \quad G_i(x)_{jk} = \frac{\partial^2 r_i(x)}{\partial x_j \partial x_k}, \quad i = 1, ..., m. \tag{1.5}$$

Con esta notación se puede reescribir la función objetivo como

$$f(x) = \frac{1}{2}\sum_{i=1}^{m}[y_i - g(x, t_i)]^2 = \frac{1}{2}\sum_{i=1}^{m} r_i^2(x) = \frac{1}{2}\|r(x)\|_2^2. \tag{1.6}$$

El gradiente y la matriz hessiana de esta función objetivo, $f(x)$, se pueden expresar en términos de las matrices jacobiana $J(x)$ y hessiana $G(x)$ de la siguiente manera:

$$\nabla f(x) = \sum_{i=1}^{m} r_i(x)\nabla r_i(x) = J(x)^T r(x), \tag{1.7}$$

$$\begin{aligned} \nabla^2 f(x) &= \sum_{i=1}^{m} \nabla r_i(x)\nabla r_i(x)^T + \sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x) = \\ &= J(x)^T J(x) + Q(x), \end{aligned} \tag{1.8}$$

donde $Q(x) = \sum_{i=1}^{m} r_i(x)G_i(x)$.

En la mayoría de los problemas, las derivadas de los residuos y por ende el cálculo de la matriz jacobiana $J(x)$ es relativamente sencillo pudiendo así deducir el gradiente de $f(x)$ (1.7) con un pequeño coste operativo. Por otra parte, si nos fijamos en la hessiana (1.8) se distinguen dos términos. El primero se puede obtener sin necesidad de calcular ninguna derivada segunda de los residuos, por lo que se deduce fácilmente usando la matriz jacobiana previamente calculada. Esto es una de las principales características de las que se beneficia el problema de mínimos cuadrados. Además, este término generalmente es mucho más importante que el segundo, ya sea porque los residuos $r_i(x)$ son pequeños, siendo esto lo esperado al buscar la mejor aproximación a la solución real, o porque $\nabla^2 r_i(x)$ es pequeño.

Ejemplo 1.3. En un estudio del efecto que tiene una cierta medicación en pacientes, se han registrado distintas muestras de sangre cada cierto intervalo de tiempo tras el suministro de una dosis. Mediante este proceso podemos saber la concentración de medicación en las muestras, y_i, dependiendo del tiempo que haya pasado desde su administración, t_i. Basándonos en los resultados, queremos encontrar los parámetros $x = (x_1, x_2, x_3, x_4, x_5)$ que proporcionen una mejor predicción de la concentración en función del tiempo usando la función

$$g(x; t) = x_1 + tx_2 + t^2 x_3 + x_4 e^{-x_5 t}. \tag{1.9}$$

Una manera de reducir las discrepancias entre los valores experimentalmente obtenidos y los calculados al imponer a $g(x; t)$ unos determinados tiempos (los experimentales) es formular el correspondiente problema de mínimos cuadrados, donde la función objetivo a minimizar era

$$f(x) = \frac{1}{2} \sum_{i=1}^{m} [y_i - g(x, t_i)]^2.$$

Gráficamente, si representamos los resultados experimentales y los obtenidos al usar el modelo, una vez encontrado el vector x deseado, se pueden observar los distintos residuos, indicados en la siguiente figura con líneas verticales punteadas.

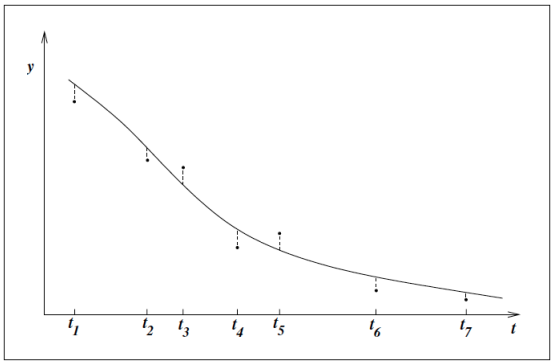

Figura 1.1: Discrepancias encontradas entre los datos experimentales recogidos (puntos) y el modelo matemático obtenido al realizar el ajuste de mínimos cuadrados (línea continua)

El modelo anterior es un ejemplo del llamado por los estadísticos modelo de regresión donde se considera que los tiempos se conocen con bastante precisión pero que las concentraciones y_i pueden contener errores debido a las limitaciones en la técnica de recogida de muestras o en los equipos de medida.

1.2. Punto de vista estadístico

Ahora analizaremos las motivaciones estadísticas por las que es adecuado elegir el método de mínimos cuadrados. Sabemos que las diferencias entre el modelo y las observaciones se representa por

$$r_i(x) = g(x; t_i) - y_i. \tag{1.10}$$

Parece razonable suponer que estos residuos son independientes e igualmente distribuidos con una varianza σ^2 y una función de densidad de probabilidad $f_\sigma(\cdot)$. Por lo tanto, la probabilidad de un conjunto de observaciones y_i con $i = 1, 2, \ldots, m$ vendrá dada por la función de densidad conjunta

$$\phi(y; x, \sigma) = \prod_{i=1}^{m} f_\sigma(r_i(x)). \tag{1.11}$$

Analizando la fórmula anterior, se puede deducir que el valor más probable de x se obtiene al maximizar $\phi(y; x, \sigma)$ con respecto a x considerando los valores observados y_i fijos.

Observación 1.4. En la práctica se suele utilizar el logaritmo de esta función

$$\widehat{\phi}(y; x, \sigma) = ln(\phi(y; x, \sigma)) = \sum_{i=1}^{m} ln(f_\sigma(r_i(x))) \tag{1.12}$$

por lo que el vector x más probable, denotado por x^*, será

$$x^* = \arg\max_{x \in X} \widehat{\phi}(y; x, \sigma). \tag{1.13}$$

A este valor se le denomina estimador de máxima verosimilitud.

Cuando se pueda suponer que las discrepancias siguen una distribución normal tendremos

$$f_\sigma(r_i) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{r_i^2}{2\sigma^2}), \tag{1.14}$$

y sustituyendo esto en la ecuación (1.11) queda

$$\phi(y; x, \sigma) = (2\pi\sigma^2)^{-m/2} exp(-\frac{1}{2\sigma^2} \sum_{i=1}^{m} [g(x; t_i) - y_i]^2). \tag{1.15}$$

De la expresión anterior se deduce claramente que, independientemente del valor que tome la varianza σ^2, si se quiere maximizar la función de densidad conjunta es necesario minimizar la suma de los cuadrados (1.1). En resumen, cuando podamos suponer que las discrepancias son independientes e igualmente distribuidas con una función de distribución normal el estimador de máxima verosimilitud se obtendrá minimizando la suma de los cuadrados.

1.3. El problema lineal de mínimos cuadrados

1.3.1. Introducción al problema

A la hora de analizar unos datos experimentales muchas veces nos daremos cuenta de que el modelo matemático $g(x;t)$ se puede tomar como una función lineal de x. Estos casos, denominados problemas lineales de mínimos cuadrados, son los que se van a analizar en esta sección. Con el objetivo de reducir los posibles errores en la recopilación de los datos, como ya se comentó, se suele coger un número bastante mayor al de los parámetros desconocidos del modelo. Por este motivo, se puede decir de forma general que al querer hallar los valores de los parámetros nos encontramos ante un sistema de ecuaciones sobredeterminado.

Inicialmente haremos un planteamiento más genérico desde un punto de vista matricial. Dada una matriz $A \in \mathbb{R}^{m\times n}$ y un vector $y \in \mathbb{R}^m$ queremos encontrar el vector $x \in \mathbb{R}^n$ que haga mínima la norma residual $\|Ax - y\|_2$, es decir, el vector x para el cual Ax se aproxime lo máximo posible a y.

Observación 1.5. Cabe señalar que, de forma general, sabemos que para que $Ax = y$ tenga al menos una solución el término independiente y debe ser combinación lineal de los vectores columna de A. Además, para que esta solución sea única los vectores columna deben ser independientes, formando una base del subespacio imagen. Una interpretación geométrica de la matriz A sería verla como la matriz de una aplicación de $\mathbb{R}^n$ en $\mathbb{R}^m$. La existencia de solución para todo $y \in \mathbb{R}^m$ exige la sobreyectividad de la aplicación, $n \geq m$, y la unicidad de la misma su inyectividad, es decir, que $m = n$ y además que el determinante de la matriz A sea distinto de cero.

Como el problema que nos ocupa es sobredeterminado, pues poseemos más datos experimentales que parámetros desconocidos, tendremos $m > n$ luego se puede suponer que el sistema anteriormente propuesto, $Ax = y$, no tiene solución y nuestro objetivo será encontrar el vector x, que no es más que los parámetros desconocidos, con el que mejor se ajuste el modelo a los datos observados, es decir,

$$\min_x \|Ax - y\|_2, \qquad A \in \mathbb{R}^{m\times n}, \quad y \in \mathbb{R}^m, \tag{1.16}$$

donde $\|\cdot\|_2$ denota la norma Euclídea.

En el caso de que $\text{rang}(A) < n$ la solución que minimize las discrepancias entre Ax e y no va a ser única pero entre las múltiples soluciones se tomará aquella que tenga norma Euclídea mínima.

Observación 1.6. En verdad, existen múltiple caminos para medir las discrepancias entre los datos observados y el modelo. La forma más habitual es la planteada en el problema lineal de mínimos cuadrados, por razones estadísticas que ya se

vieron en la Sección 1.2. Sin embargo, no es la única. Otras medidas comunes son calcular el máximo valor absoluto

$$\max_{j=1,2,\ldots,m} \mid r_j \mid,$$

o la suma de valores absolutos

$$\sum_{j=1}^{m} \mid r_j \mid,$$

que corresponderían a minimizar la norma infinito ($\|\cdot\|_\infty$) y la norma ℓ_1 ($\|\cdot\|_1$) respectivamente.

1.3.2. El problema lineal de mínimos cuadrados

Definición 1.7. Las funciones $r_i(x)$ en estos problemas son lineales en x, pudiendo escribirse el **vector residual**, desde un punto de vista matricial, como

$$r(x) = Ax - y, \tag{1.17}$$

donde la matriz $A \in \mathbb{R}^{mxn}$ y el vector $y \in \mathbb{R}^m$, ambos siendo independientes de x.

Por lo tanto, se puede describir el problema lineal de mínimos cuadrados como aquel que busca minimizar la suma de los cuadrados de los residuos, teniendo como función objetivo

$$f(x) = \frac{1}{2}\|Ax - y\|_2^2. \tag{1.18}$$

El gradiente y la matriz hessiana de la función anterior son

$$\nabla f(x) = A^T(Ax - y), \tag{1.19}$$

$$\nabla^2 f(x) = A^T A. \tag{1.20}$$

Comparando los resultados con los obtenidos al analizar el problema general (1.7) y (1.8), podemos notar que el segundo término que aparece en $\nabla^2 f(x)$ en el caso general, desaparece en (1.20) porque $\nabla^2 r_i = 0$ para todo $i = 1, \ldots, m$ en estos problemas.

Por otra parte, se puede comprobar que $f(x)$ en los problemas lineales es convexa, propiedad que no necesariamente se cumple en los problemas no lineales. Con esto se puede concluir, gracias al siguiente teorema, que el vector x^* que verifique $\nabla f(x^*) = 0$ no solo es un mínimo local de $f(x)$ sino que es un mínimo global.

Teorema 1.8. *Sea $f(x)$ una función convexa. Cualquier x^* que sea mínimo local es también mínimo global.*

Demostración. Razonemos por reducción al absurdo. Supongamos que x^* es un mínimo local pero no global. Entonces al no ser global existirá un punto $z \in \mathbb{R}^n$ tal que $f(z) < f(x^*)$. Tomamos ahora el segmento que une z con x^*

$$x = \lambda z + (1 - \lambda)x^*, \tag{1.21}$$

donde $\lambda \in (0, 1]$.

Por la convexidad de f se tiene

$$f(x) \leq \lambda f(z) + (1 - \lambda)f(x^*) < f(x^*) \tag{1.22}$$

llegando a contradicción pues entonces cualquier punto x cercano a x^* que este en el segmento (1.21) verificará la desigualdad estricta anterior lo que implicaría que x^* no es mínimo local. □

Proposición 1.9. Usando la condición de mínimo $\nabla f(x^*) = 0$ y (1.19) se deduce que x^* debe ser solución del siguiente sistema lineal de ecuaciones conocido por **ecuaciones normales**

$$A^T Ax = A^T y. \tag{1.23}$$

Para resolver este sistema se han desarrollado diversos algoritmos entre los cuales vamos a explicar superficialmente tres de los más utilizados: la factorización de Cholesky, la factorización QR y la descomposición en valores singulares (SVD). En los tres supondremos que $m \geq n$ y que $\text{rang}(A) = n$.

1.3.3. Factorización de Cholesky

La forma más directa de abordar el problema es construir la matriz $A^T A$ y resolver las ecuaciones normales (1.23). Para ello, recordamos el siguiente resultado.

Teorema 1.10. *Sea A una matriz real y regular. Se cumple que*

(a) Existe una matriz triangular inferior e invertible R tal que $A = RR^T$ si, y sólo si A es simétrica y definida positiva.

(b) Además, si los elementos de la diagonal de R son positivos entonces la factorización será única. Esta es la denominada factorización de Cholesky.

La demostración del teorema anterior se puede encontrar en [7] p.163 y en [16] p.33.

Volviendo a las ecuaciones normales (1.23), como $A^T A$ es una matriz simétrica y definida positiva, el Teorema 1.10 garantiza que existe una única matriz triangular inferior $R \in \mathbb{R}^{n \times n}$ con elementos diagonales positivos que verifica

$$A^T A = RR^T. \tag{1.24}$$

Luego (1.24) es la **factorización de Cholesky** de $A^T A$.

El primer procedimiento para resolver las ecuaciones normales (1.23) consistirá en los siguientes pasos:

1. Calcular la matriz A^TA y el término independiente A^Ty .
2. Hallar la factorización de Cholesky de la matriz A^TA.
3. Resolver el sistema $RR^Tx = A^Ty$, que no es más que realizar dos sustituciones, una progresiva y otra regresiva, ya que R es una matriz triangular inferior
$$Rz = A^Ty, \quad R^Tx = z.$$

Este método es muy usado en la práctica y generalmente es eficiente. Su gran ventaja es el pequeño coste operativo. Sin embargo, posee una gran desventaja pues el número de condición de A^TA se comporta como el cuadrado del de A. Como el error relativo de una solución es proporcional al número de condición, esto provoca que con este método se obtengan soluciones menos precisas. Además, en los casos en los que A esté mal condicionada, es decir, que cambios relativamente pequeños en los datos provoquen cambios relativamente grandes en la solución, esta factorización podría incluso dar lugar a la aparición de elementos negativos en la diagonal de R.

1.3.4. Factorización QR

El segundo procedimiento proporciona la solución de (1.23) sin construir explícitamente la matriz A^TA. Para ello se tiene en cuenta en primer lugar, que la norma euclídea es invariante frente a transformaciones ortogonales, es decir,

$$\min_x \|Ax - y\|_2 = \min_x \|Q^T(Ax - y)\|_2 = \min_x \|Q^TAx - Q^Ty\|_2, \tag{1.25}$$

si $Q \in \mathbb{R}^{m\times m}$ es una matriz ortogonal, es decir, $Q^T = Q^{-1}$.

Partiendo de lo anterior, la idea de este segundo procedimiento, es encontrar una matriz Q ortogonal tal que $Q^TA = R$ donde R sea una matriz triangular en un sentido "amplio", pues de forma general esta matriz R no va a ser cuadrada. El esquema deseado sería:

$$Q^TA = R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \begin{matrix} \updownarrow n \\ \updownarrow m-n \end{matrix} \qquad \text{y} \qquad Q^Ty = \begin{pmatrix} c \\ d \end{pmatrix} \begin{matrix} \updownarrow n \\ \updownarrow m-n \end{matrix} \tag{1.26}$$

donde $R_1 \in \mathbb{R}^{n\times n}$ es triangular superior y regular (pues tiene el mismo rango que A).

Teorema 1.11. *Dada una matriz $A \in \mathbb{R}^{m\times n}$ con $m \geq n$ y rang$(A) = n$, existe una matriz $Q \in \mathbb{R}^{mxm}$ ortogonal y una matriz $R \in \mathbb{R}^{mxn}$ triangular superior en sentido amplio tal que*
$$A = QR \tag{1.27}$$

Esta factorización no es única y existen distintas formas de construirla como usar las tranformaciones de Householder o la ortonormalización de Gram-Schmidt. El desarrollo de estos métodos no se expondrá en este TFG pero se pueden encontrar fácilmente en el Capítulo 2 de [1].

Utilizando el Teorema 1.11 que proporciona la factorización QR de la matriz A en (1.18), se tiene

$$\|Ax - y\|_2^2 = \|Q^T Ax - Q^T y\|_2^2 = \|R_1 x - c\|_2^2 + \|d\|_2^2, \tag{1.28}$$

donde R_1, c y d son como en (1.26). Lo primero que se observa, es que no es necesario el cálculo completo de la factorización QR. Es suficiente calcular las matrices R_1 y $Q_1^T y = c$ siendo Q_1 la submatriz de Q formada por sus primeras n columnas, y realizar a continuación una sustitución regresiva para resolver el sistema triangular $R_1 x = c$. Dicho de otra forma

$$x^* = {R_1}^{-1} Q_1^T y. \tag{1.29}$$

Además, de (1.28) también se deduce que la magnitud del error cometido, al quedar anulado el primer término, es $\|d\|_2^2$.

Este método basado en la factorización QR tiene la ventaja de que al no construir $A^T A$ no degrada el número de condición como ocurría en el caso de la factorización de Cholesky. Sin embargo, en algunas situaciones es necesario un método más robusto que posea una mayor información sobre la sensibilidad de la solución frente a perturbaciones en los datos. Por este motivo desarrollaremos en el siguiente apartado un tercer procedimiento basado en la SVD.

1.3.5. Descomposición de valores singulares (SVD)

Vamos a utilizar el siguiente teorema cuya demostración puede encontrarse en el Capítulo 1 de [1].

Teorema 1.12. *Sea $A \in \mathbb{R}^{m\times n}$. Existen matrices ortogonales $U \in \mathbb{R}^{m\times m}$ y $V \in \mathbb{R}^{n\times n}$ tales que*

$$U^T A V = \Sigma, \tag{1.30}$$

donde Σ es una matriz diagonal en sentido amplio, es decir,

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \sigma_n \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{matrix} \uparrow \\ \\ n \\ \\ \downarrow \\ \uparrow \\ m-n \\ \downarrow \end{matrix} \tag{1.31}$$

con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$.

Denotaremos con u_i a la columna i-ésima de la matriz U y con v_i a la columna i-ésima de la matriz V.

Definición 1.13. Se llaman **valores singulares** de A a las raíces cuadradas positivas de los autovalores de A^TA. Estos se representan por σ_i.

Corolario 1.14. *En la descomposición en valores singulares si* $rang(A) = r < n$ *se verificará*

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_n = 0 \tag{1.32}$$

En nuestro caso, volviendo a (1.23), la descomposición quedaría

$$A^TA = V\Sigma^2V^T \tag{1.33}$$

donde las columnas de V son los autovectores de A^TA con autovalores σ_i^2, $1 \leq i \leq n$.

A la hora de minimizar (1.18) tendremos, usando que la norma euclídea es invariante frente a transformaciones ortogonales y tomando $w = V^Tx \in \mathbb{R}^n$,

$$\begin{aligned} \|Ax - y\|_2^2 = \|U^TAVV^Tx - U^Ty\|_2^2 = \|\Sigma w - U^Ty\|_2^2 = \\ = \sum_{i=1}^{r}(\sigma_i w_i - u_i^Ty)^2 + \sum_{i=r+1}^{m}(u_i^Ty)^2. \end{aligned}$$

Luego el valor x óptimo será

$$x^* = \sum_{i=1}^{r}\frac{u_i^Ty}{\sigma_i}v_i, \tag{1.34}$$

y el error mínimo

$$\sum_{i=r+1}^{m}(u_i^Ty)^2. \tag{1.35}$$

Con este método poseemos una gran información sobre la sensibilidad frente a perturbaciones. Si nos fijamos en la fórmula (1.34), sabemos que si σ_i es próximo a cero entonces una pequeña perturbación en el valor de y afectará mucho al valor final de x^* . Esta información es especialmente útil cuando A esté mal condicionada y los últimos valores singulares sean relativamente pequeños comparados con el valor de σ_1 ($\sigma_r/\sigma_1 \ll 1$). Una solución aproximada, que sería menos sensible a estas perturbaciones, sería omitir los últimos términos en el sumatorio de (1.34).

Definición 1.15. Utilizando la descomposición en valores singulares, se define la **pseudoinversa** de una matriz A como

$$A^+ = V\Sigma^+ U^T, \tag{1.36}$$

donde $\Sigma^+ = diag(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0) \in \mathbb{R}^{n\times m}$, y $A = U\Sigma V^T$ es la SVD de A.

Una explicación más detallada sobre la pseudoinversa de una matriz y sus propiedades se puede encontrar en el Apéndice A.

Con esta nueva definición se puede reescribir el resultado (1.34) como

$$x^* = A^+ y. \tag{1.37}$$

En resumen, según el modelo matemático que se quiera ajustar siempre y cuando este sea lineal tendrá un método, de los explicados anteriormente, más adecuado que los otros. En el caso en que $m \gg n$ será más recomendable el método basado en la factorización de Cholesky, siendo el más económico computacionalmente hablando. Sin embargo, este método no es útil cuando la matriz A sea de rango deficiente o esté mal condicionada. En estos casos es recomendable utilizar la factorización QR. Por último, los casos en los que A sea de rango deficiente y/o muy sensible a pequeñas perturbaciones es preferible usar el método más robusto, basado en la SVD, aunque esto implique un mayor coste computacional.

Capítulo 2

El problema no lineal de mínimos cuadrados

En este capítulo se discutirán distintos métodos para la solución numérica de los problemas no lineales de mínimos cuadrados. Por lo general, los métodos que se usan para resolver estos problemas son iterativos, y cada iteración requiere solucionar un problema lineal de mínimos cuadrados, para lo que se utilizarán los métodos ya tratados en el capítulo anterior.

Por otra parte, los problemas no lineales de mínimos cuadrados se pueden ver como resolución de un sistema de ecuaciones no lineales con más ecuaciones que incógnitas, y se les puede clasificar como un caso especial dentro de los problemas de optimización en $\mathbb{R}^n$. Analizaremos primero cómo reconocer un mínimo y luego expondremos distintos métodos numéricos de resolución, analizando las direcciones de búsqueda y convergencia en cada caso.

2.1. Condiciones para reconocer los mínimos

El objetivo de nuestro problema no lineal de mínimos cuadrados es encontrar el mínimo global de la suma de cuadrados de m funciones no lineales, es decir,

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{con} \quad f(x) = \frac{1}{2}\sum_{i=1}^{m} r_i^2(x), \quad m \geq n, \tag{2.1}$$

donde cada $r_i(x)$ corresponde a una función no lineal definida en $\mathbb{R}^n$.

La estructura de este problema ya se desarrolló en la Sección 1.1. El caso en el que $m = n$ es un caso especial de solución de un sistema de ecuaciones no lineales que no vamos a tratar, por lo que consideraremos que $m > n$. Además, supondremos que todas las funciones $r_i(x)$ son al menos de clase $\mathscr{C}^2$.

Nuestra primera preocupación, por lo tanto, es conocer las condiciones necesarias y suficientes para que x pueda ser considerado un mínimo de la función $f(x)$. Para ello, empezamos recordando unos conceptos y teoremas básicos aplicables a cualquier función f.

Definición 2.1. Sea $n \geq 1$ y $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ una función cualquiera. Diremos que $x^* \in \mathbb{R}^n$ es:

- Un **mínimo global** de f si $f(x^*) \leq f(x) \quad \forall x \in \mathbb{R}$.
- Un **mínimo local** de f si existe un entorno $\mathscr{U}$ de x^* tal que $f(x^*) \leq f(x)$ $\forall x \in \mathscr{U}$ (a veces es denominado mínimo local débil).
- Un **mínimo local estricto** o mínimo local fuerte de f si existe un entorno $\mathscr{U}$ de x^* tal que $f(x^*) < f(x) \quad \forall x \in \mathscr{U}$ con $x \neq x^*$.

Cuando la función f que estemos analizando sea 2 veces continuamente diferenciable podremos localizar los mínimos locales x^* analizando su gradiente y su matriz hessiana. Recordemos cuales son las condiciones necesarias y/o suficientes que debe cumplir x^* para que sea mínimo local.

Teorema 2.2.
Condiciones necesarias de primer orden y segundo orden:

Si x^* es un mínimo local y f es continuamente diferenciable en un entorno abierto $\mathscr{U}$ de x^* entonces

$$\nabla f(x^*) = J(x^*)^T r(x^*) = 0, \tag{2.2}$$

donde x^* es denominado punto crítico.

Si además $\nabla^2 f$ existe y es continua en $\mathscr{U}$ entonces $\nabla^2 f(x^*)$ es semidefinida positiva.

Condiciones suficientes de segundo orden:

Si $\nabla f(x^*) = 0$ y $\nabla^2 f$ es continua en un entorno abierto de x^* y además $\nabla^2 f(x^*)$ es definida positiva, entonces x^* es un mínimo local estricto de f.

La demostración de este teorema se encuentra detallada en [11] pp.15-17. Además, es de señalar que las condiciones suficientes no son necesarias.

Ahora aplicamos estos resultados al caso concreto de los problemas no lineales de mínimos cuadrados. Usando la notación descrita en la Sección 1.1 y suponiendo que la matriz $J(x)$ es de rango columna completo (las columnas de $J(x)$ son linealmente independientes), podemos reescribir la matriz hessiana (1.8) como:

$$\nabla^2 f(x) = J^T J - \gamma G_w = J^T (I - \gamma (J^+)^T G_w J^+) J, \tag{2.3}$$

donde $\gamma = \|r\|_2$, $G_w = \sum_{i=1}^m w_i G_i$ y $w = r/\gamma$.

La deducción de la ecuación (2.3) se obtiene fácilmente usando la propiedad de la matriz pseudoinversa $J^+(x)J(x) = I_n$ (por ser de rango columna completo).

Por otra parte, podemos ver el problema de minimización de $f(x)$ como el de encontrar el punto en la superficie n-dimensional $z = r(x)$ más cercano al origen.

Definición 2.3. Se define la **matriz normal de curvatura** de la superficie $z = r(x)$ con respecto al vector normal w como

$$K = (J^+)^T G_w J^+. \tag{2.4}$$

Esta matriz es simétrica, puesto que G_w es simétrica. Vamos a denotar a sus autovalores ordenados como

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n. \tag{2.5}$$

A los inversos no nulos de estos autovalores de K se les denomina radios principales de curvatura de la superficie, con respecto al vector normal w,

$$\sigma_i = \frac{1}{\lambda_i} \quad \text{con} \quad \lambda_i \neq 0. \tag{2.6}$$

Usando las definiciones y teoremas anteriores junto a que $J(x^*)$ es de rango columna completo y que

$$\nabla^2 f(x^*) = J^T(I - \gamma K)J \tag{2.7}$$

se deduce lo siguiente.

Proposición 2.4. *$\nabla^2 f(x^*)$ es definida positiva si, y sólo si $I - \gamma K$ es definida positiva en x^*. Además, si se cumple alguna de las condiciones anteriores siendo x^* un punto crítico entonces x^* será un mínimo local.*

Demostración. Tomando $y = Jx$ podemos deducir

$$x^T \nabla^2 f(x^*) x = x^T J^T (I - \gamma K) J x = (Jx)^T (I - \gamma K)(Jx) = y^T (I - \gamma K) y.$$

Por otra parte, $x = 0 \Leftrightarrow y = 0$ pues hemos tomado $y = Jx$ donde J es una matriz con columnas independientes. Con esto y usando la definición de matriz definida positiva queda probada la doble implicación.
Por último, el caso en que x^* sea un punto crítico se deduce aplicando directamente el Teorema 2.2. □

Corolario 2.5. *Sea x^* un punto crítico.*

- *Si $1 - \gamma\lambda_1 > 0$ en x^*, entonces x^* es un mínimo local.*
- *Si $1 - \gamma\lambda_n < 0$ en x^*, entonces x^* es un máximo local.*
- *Si $1 - \gamma\lambda_1 \leq 0 \leq 1 - \gamma\lambda_n$ en x^*, entonces x^* es un punto de silla.*

En el caso de ajustar un modelo a unos datos experimentales se considera la superficie

$$z = (g(x, t_1), \ldots, g(x, t_m))^T, \tag{2.8}$$

y el objetivo del problema es encontrar el punto en la superficie más cercano al vector observado $y \in \mathbb{R}^m$.

2.2. Metodología general

Como ya dijimos al comienzo del capítulo, los métodos que utilizaremos para resolver los problemas no lineales de mínimos cuadrados son iterativos, siendo necesario partir de un iterante inicial que denotaremos por x_0. Éste en muchos casos debe ser elegido con cuidado, dependiendo del algoritmo que vayamos a aplicar, si queremos alcanzar una cierta convergencia. A partir de este valor inicial, los algoritmos generarán una sucesión $\{x_k\}_{k=0}^{\infty}$ que terminará cuando el algoritmo no progrese más o cuando el punto encontrado x^* cumpla con todas las restricciones de tolerancia impuestas y se considere una aproximación suficientemente precisa.

Para decidir cómo pasar de un iterante x_k al siguiente x_{k+1} los algoritmos extraerán información de la función objetivo f y de sus derivadas en x_k, y a veces incluso en los iterantes previos $(x_0, \ldots x_{k-1})$, con el objetivo de disminuir la suma de los cuadrados de los residuos.

2.2.1. Dos estrategias básicas

Existen dos estrategias fundamentales que siguen la mayoría de los algoritmos para construir la sucesión $\{x_k\}_k$.

• Búsqueda de línea

El algoritmo inicialmente elige una dirección p_k, que se denomina **dirección de búsqueda**. Ésta marcará la dirección que hay que tomar desde el iterante actual para encontrar el siguiente. La distancia que hay que moverse en dicha dirección para encontrar la máxima minimización de la función objetivo se puede calcular resolviendo

$$\min_{\alpha>0} f(x_k + \alpha p_k). \tag{2.9}$$

Esta distancia α_k se denomina **longitud de paso**.
Generalmente, encontrar el valor que consigan el mínimo exacto de (2.9) es muy costoso, por lo que se suelen usar aproximaciones.

Una vez calculado el siguiente iterante x_{k+1} se repite el proceso con una nueva dirección de búsqueda.

• **Región de confianza**

Por su parte, los métodos basados en la región de confianza construyen una función m_k, cuyo comportamiento es muy similar al de f en las proximidades del iterante actual x_k, aunque puede diferir de f para valores lejanos. Una vez construida m_k buscaremos el valor de p_k que la minimize, siempre y cuando este mínimo esté dentro de la región permitida para la aproximación (la región en que m_k y f se parecen), es decir, de la denominada **región de confianza** (un disco n-dimensional centrado en x_k). Por lo tanto, para encontrar p_k será necesario resolver

$$\min_p m_k(x_k + p) \quad \text{con } \|p\|_2 \leq \delta_k. \tag{2.10}$$

En (2.10), se hace uso del **radio de la región de confianza** $\delta_k > 0$, para garantizar que $x_k + p_k$ esté dentro de la región de confianza. De forma general, y es lo que hemos hecho anteriormente, la región de confianza se elige con forma circular aunque también podría ser elipsoidal o cuadrada.

En los casos en que la solución obtenida no produzca un decrecimiento suficiente de la función f será necesario reducir las dimensiones de la región de confianza (en el caso circular su radio) y resolver de nuevo (2.10).

La principal diferencia entre las dos estrategias que acabamos de describir es que en la primera se fija una dirección de búsqueda p_k y luego se trata de encontrar la distancia óptima (longitud de paso α_k) y sin embargo, en la segunda se fija en primer lugar la distancia máxima (en el caso de que sea una región circular se fija el radio δ_k) y luego se busca la dirección y longitud óptimas dentro de esa región, reduciendo la distancia inicial en el caso de no encontrar ninguna mejora.

2.2.2. Direcciones de búsqueda

El principal objetivo de las direcciones de búsqueda es conseguir un descenso en el valor de la función objetivo, es decir, conseguir que $f(x_k + \alpha_k p_k) < f(x_k)$ para un valor α_k positivo suficientemente pequeño. Una condición que garantiza este descenso es

$$\nabla f(x_k)^T p_k < 0. \tag{2.11}$$

La mayoría de los algoritmos que buscan una solución para el problema de mínimos cuadrados se basan en encontrar unos valores de p_k y α_k que aseguren una disminución suficiente de la función objetivo f.

Vamos a presentar distintas formas que se usan para elegir la dirección de búsqueda p_k, pero antes enunciaremos un teorema que será de utilidad más adelante.

Teorema 2.6. *Sea $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ una función diferenciable, $p \in \mathbb{R}^n$ y $\alpha \in \mathbb{R}$. Se tiene que*

$$f(x + \alpha p) = f(x) + \alpha p^T \nabla f(x + tp), \tag{2.12}$$

para algún $t \in (0, \alpha)$.

Además, si $f \in \mathscr{C}^2$ se cumple

$$f(x + \alpha p) = f(x) + \alpha p^T \nabla f(x) + \frac{1}{2}\alpha^2 p^T \nabla^2 f(x + tp)p, \tag{2.13}$$

para algún $t \in (0, \alpha)$.

Las direcciones de búsqueda más usadas son:

❖ Dirección de máximo descenso

Es la que consigue que la función objetivo f disminuya más rápidamente de una iteración a la siguiente. Consiste en tomar

$$p_k = -\nabla f(x_k). \tag{2.14}$$

Si analizamos el Teorema 2.6 vemos que el incremento de f viene dado por $p_k^T \nabla f(x_k)$. Como $p_k^T \nabla f(x_k) = \|p_k\|_2 \|\nabla f(x_k)\|_2 \cos\theta$, donde θ es el ángulo entre p_k y $\nabla f(x_k)$, si tomamos $\cos\theta = -1$ obtendremos el mínimo deseado con p_k dado por (2.14).

La gran ventaja de esta dirección es que no necesita el cálculo de segundas derivadas. Sin embargo, en problemas complejos puede presentar una convergencia muy lenta.

❖ Dirección de Newton

Esta dirección es de las más utilizadas. Se deduce de la aproximación de Taylor de segundo orden

$$f(x_k + p_k) \approx f(x_k) + p_k^T \nabla f(x_k) + \frac{1}{2} p_k^T \nabla^2 f(x_k) p_k. \tag{2.15}$$

Si vemos el lado derecho de (2.15) como función de p_k e igualamos su gradiente a cero y suponemos que $\nabla^2 f(x_k)$ es definida positiva entonces

$$p_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k). \tag{2.16}$$

En este caso se dice que la longitud de paso es 1.

Esta dirección proporciona al algoritmo una rápida convergencia aunque sea bastante costosa de obtener, pues requiere calcular las derivadas segundas.
Además, (2.16) no siempre está bien definida pues puede ocurrir que $\nabla^2 f(x_k)$ no sea definida positiva o que no se cumpla la condición de descenso.

❖ **Dirección de Quasi-Newton**

Es una mejora de la dirección de Newton en el sentido de que no va a ser necesario el cálculo de la matriz hessiana y aún así se puede seguir consiguiendo una convergencia superlineal. La idea consiste en sustituir la matriz hessiana $\nabla^2 f(x_k)$ por una aproximación que denotaremos por B_k. Esta aproximación se definirá con detalle en la Sección 2.5.2. La dirección de búsqueda, por tanto, queda definida como

$$p_k = -B_k^{-1} \nabla f(x_k). \tag{2.17}$$

❖ **Dirección de gradiente conjugado**

En este caso, la dirección de búsqueda tomará la forma

$$p_k = -\nabla f(x_k) + a_k p_{k-1}, \tag{2.18}$$

donde a_k es un escalar con el que se asegura que p_k y p_{k-1} sean conjugados, es decir, $p_{k-1}^T p_k = 0$. Como se observa en (2.19), la dirección de búsqueda es una combinación lineal de la dirección de máximo descenso y de la dirección de búsqueda utilizada en la iteración anterior.

❖ **Dirección para métodos con región de confianza**

En los métodos que usan una región de confianza (2.10) generalmente se toma la función aproximada m_k como

$$m_k(x_k + p_k) = f(x_k) + p_k^T \nabla f(x_k) + \frac{1}{2} p_k^T B_k p_k \tag{2.19}$$

donde B_k puede ser la matriz hessiana $\nabla^2 f(x_k)$ o alguna aproximación de esta. Dependiendo del valor que le demos a esta matriz podemos obtener distintas direcciones de búsqueda.

- Si $B_k = 0$ entonces el problema se reduce a

  $$\text{mín } f(x_k) + p_k^T \nabla f(x_k) \quad \text{ sujeto a } \quad \|p_k\|_2 \leq \delta_k. \tag{2.20}$$

  La solución a este problema es de la forma $p_k = -\delta_k \nabla f(x_k)/\|\nabla f(x_k)\|_2$, que no es más que la dirección de máximo descenso modulada por los distintos radios de las regiones de confianza, δ_k.

- Si $B_k = \nabla^2 f(x_k)$ entonces la función cuadrática m_k toma la misma forma que en el caso del método de Newton (2.16). Este método posee una gran ventaja frente al anterior, pues gracias a la restricción de la región de confianza $\|p_k\|_2 < \delta_k$, no será necesaria la condición de que $\nabla^2 f(x_k)$ sea definida positiva para que exista la dirección de búsqueda p_k.

- Otra opción sería tomar B_k como la aproximación de la matriz hessiana $\nabla^2 f(x_k)$ utilizada en el método de Quasi-Newton, dando lugar al método de Quasi-Newton con región de confianza.

2.2.3. Longitud de paso

Una vez elegida la dirección de búsqueda, p_k, es necesario elegir la longitud de paso, α_k. En primer lugar, hay que tener en cuenta que va a ser necesario hacer un balance entre la reducción en la función objetivo f que nos gustaría alcanzar al elegir α_k y el tiempo que vamos a necesitar para alcanzar dicho valor. Generalmente, los algoritmos consiguen longitudes de paso inexactas, pues son mejorables, pero con las que se pueda alcanzar una considerable reducción de la función f a un coste operativo no muy elevado. Normalmente su cálculo tienen dos fases, una inicial de horquillado, en el que se localizan intervalos que contengan longitudes de paso deseables, y una segunda fase de interpolación, en la que se calcula una buena longitud de paso dentro de algún intervalo de los localizados en la fase previa.

Veamos algunas de las condiciones que nos ayudan a elegir α_k.

✦ Condición de Armijo
La longitud de paso α_k debe cumplir que

$$\Phi(\alpha_k) = f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k p_k^T \nabla f(x_k), \tag{2.21}$$

para algún $c_1 \in (0, 1)$.

Esto implica que la redución de f debe ser proporcional a la longitud de paso α_k y a la derivada direccional $p_k^T \nabla f(x_k) = \Phi'(0)$. El valor que toma c_1 suele ser bastante pequeño (del orden de 10^{-4}).

El efecto de utilizar esta condición se puede ver en la siguiente imagen, tomada de [11], donde $l(\alpha) = f(x_k) + c_1 \alpha p_k^T \nabla f(x_k)$.

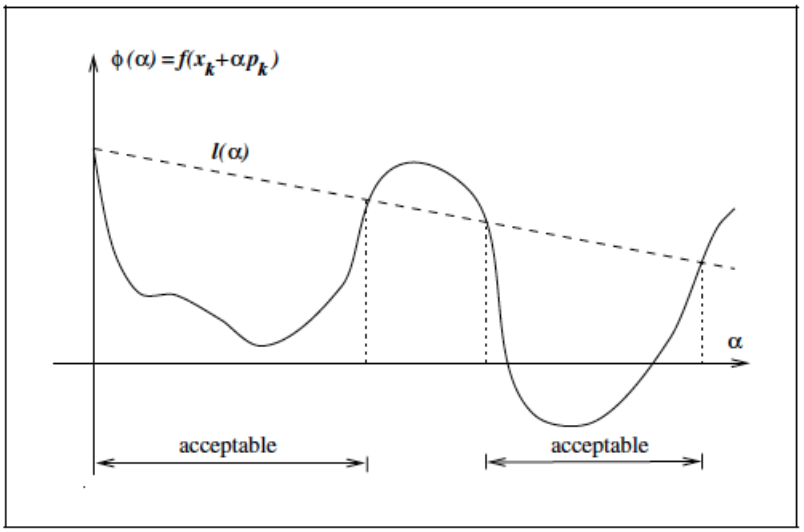

Figura 2.1: Longitudes de paso que cumplen la condición de Armijo

Sin embargo, esta condición no es suficiente para asegurar que el algoritmo vaya a progresar lo suficiente. De hecho, como puede observarse en la Figura 2.1, existen valores muy pequeños de α_k que verifican la condición. Para eliminar estos casos no deseados introducimos una segunda condición.

✦ Condición de Curvatura
La longitud de paso α_k tiene que cumplir

$$p_k^T \nabla f(x_k + \alpha_k p_k) \geq c_2 p_k^T \nabla f(x_k), \tag{2.22}$$

para algún $c_2 \in (c_1, 1)$ donde c_1 es la constante de (2.21).

Esta condición implica que la derivada $\Phi'(\alpha_k)$, es decir, la pendiente de Φ en α_k, sea c_2 veces mayor que la pendiente inicial $\Phi'(0)$.

Si combinamos las dos condiciones anteriores obtenemos lo siguiente.

✦ Condiciones de Wolfe
La longitud de paso α_k debe cumplir

$$\begin{cases} f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k p_k^T \nabla f(x_k), \\ \nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 p_k^T \nabla f(x_k), \end{cases} \tag{2.23}$$

con $0 < c_1 < c_2 < 1$.

Con estas condiciones se consigue que la función objetivo f disminuya considerablemente y que la derivada aumente lo suficiente para que x_{k+1} esté lo suficientemente separada de x_k y evitar una convergencia muy lenta del método. En la Figura 2.2 se ilustra la restricción que sobre la longitud de paso imponen las condiciones de Wolfe. También está tomada de [11].

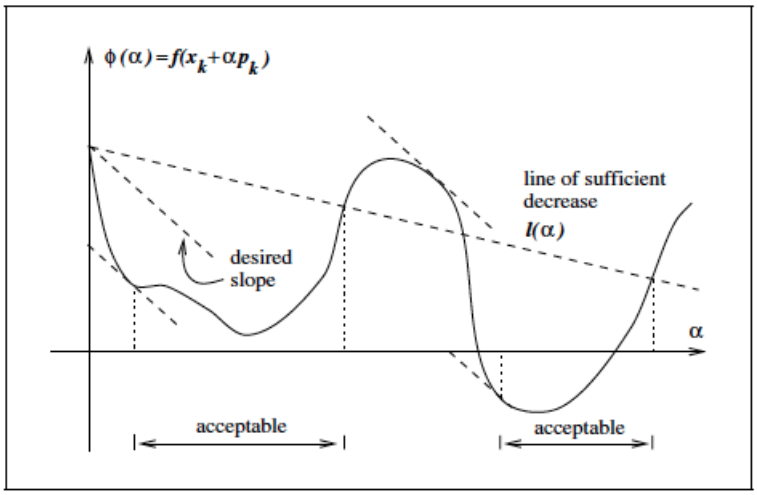

Figura 2.2: Longitudes de paso que cumplen las condiciones de Wolfe

Sin embargo, existen casos en los que aunque α_k cumpla las condiciones de Wolfe no se llega a estar cerca del mínimo deseado de Φ. Por este motivo, se suele usar una modificación de éstas, denominada **condiciones fuertes de Wolfe**, donde α_k tiene que verificar

$$\begin{cases} f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k p_k^T \nabla f(x_k), \\ |\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |p_k^T \nabla f(x_k)|, \end{cases} \tag{2.24}$$

con $0 < c_1 < c_2 < 1$.

La única diferencia con las condiciones de Wolfe (2.23) es que no se permite que la derivada $\Phi'(\alpha_k)$ sea demasiado positiva. De este modo se consigue excluir a los puntos que se encuentren muy lejos de los puntos estacionarios de Φ.

Por otra parte, como prueban Nocedal y Wright en [11] p.35, se puede asegurar el siguiente resultado.

Lema 2.7. *Sea $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ diferenciable, p_k una dirección de descenso desde x_k y f acotada inferiormente en $\{x_k + \alpha p_k \ / \alpha > 0\}$. Si $0 < c_1 < c_2 < 1$ entonces existen intervalos en los cuales el valor de α_k cumple con las condiciones de Wolfe* (2.23) *y con las condiciones fuertes de Wolfe* (2.24).

Otras condiciones, semejantes a las de Wolfe, que aseguran el éxito de encontrar una buena longitud de paso son las denominadas condiciones de Goldstein.

✦ **Condiciones de Goldstein**
Sea α_k una longitud de paso que cumple

$$f(x_k) + (1-c)\alpha_k p_k^T \nabla f(x_k) \leq f(x_k + \alpha_k p_k) \leq f(x_k) + c\alpha_k p_k^T \nabla f(x_k), \tag{2.25}$$

con $0 < c < 1/2$.

Por una parte, la primera desiguadad sirve para controlar la longitud de paso por debajo, evitando que sea demasiado pequeña, mientras que la segunda busca que se cumpla la condición de suficiente descenso. En la Figura 2.3, tomada de [11], se ilustra el efecto de imponer estas condiciones.

Las condiciones de Goldstein son bastante utilizadas en los métodos de tipo Newton. Tienen una gran ventaja frente a las condiciones de Wolfe, pues ahora no es necesario calcular $\nabla f(x_k + \alpha_k p_k)$ para cada iteración, lo que suele ser muy costoso en la práctica. Sin embargo, también poseen una gran desventaja, la primera desigualdad en (2.25) podría excluir a todos los minimizadores de $\Phi(\alpha) = f(x_k + \alpha p_k)$. Con respecto a la convergencia, ambas condiciones tienen teorías similares.

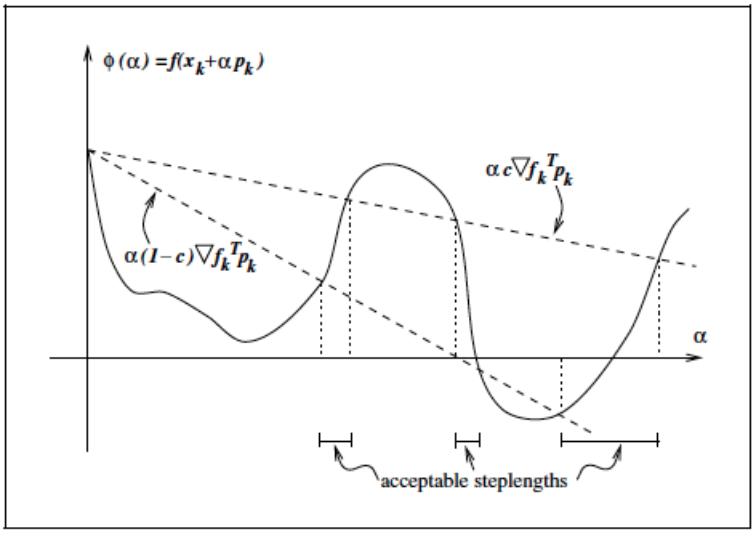

Figura 2.3: Longitudes de paso que cumplen las condiciones de Goldstein

2.3. Métodos numéricos básicos

Para resolver el problema que nos interesa, (2.1), vamos a plantearlo desde dos puntos de vista diferentes, consiguiendo distintos métodos para su solución.

En primer lugar, como ya se ha dicho antes, el problema se puede ver como un sistema de ecuaciones no lineales sobredeterminado, $r(x) = 0$, donde se va a realizar una aproximación lineal $r_c(x)$ alrededor de un punto x_c. Esto significa que

$$r_c(x) = r(x_c) + J(x_c)(x - x_c). \tag{2.26}$$

Usando esta aproximación, nuestro problema inicial se transforma en un problema lineal con la forma

$$\min_x \|r(x_c) + J(x_c)(x - x_c)\|_2, \tag{2.27}$$

que ya sabemos resolver (Ver Sección 1.3).
Esta manera de proceder, que solo va a necesitar calcular las derivadas primeras del vector residual, es la que se usa en métodos como Gauss-Newton o Levenberg-Marquardt que desarrollaremos más adelante.

La segunda forma de ver el problema (2.1) es como un caso de optimización donde se utiliza un modelo cuadrático $f_c(x)$ para tratar la función objetivo

$$f_c(x_c + z) = f(x_c) + \nabla f(x_c)^T z + \frac{1}{2} z^T \nabla^2 f(x_c) z. \tag{2.28}$$

Como el objetivo es minimizar f_c entonces igualando su derivada a cero obtenemos:

$$\nabla f(x_c)^T + \nabla^2 f(x_c)z = 0 \;\Rightarrow\; \nabla^2 f(x_c)(x_d - x_c) = -J(x_c)^T r(x_c). \tag{2.29}$$

que permite obtener $x_d = x_c + z$ como

$$x_d = x_c - (J(x_c)^T J(x_c) + Q(x_c))^{-1} J(x_c)^T r(x_c), \tag{2.30}$$

donde hemos usado las expresiones (1.7) y (1.8).
Como se puede observar, en estos casos es necesario conocer la derivada segunda del vector residual. Los métodos que siguen este planteamiento son los llamados de tipo Newton.

Es de señalar que en realidad la diferencia entre ambos planteamientos reside principalmente en si se desprecian o no las derivadas segundas del vector residual para saber si dejar o suprimir el término $Q(x_c)$. Como este término en verdad es $Q(x) = \sum_{i=1}^{m} r_i(x)G_i(x)$, en los casos en que todos los residuos $r_i(x_c)$ sean pequeños o en los que la no linealidad de los residuos en x_c sea pequeña se podrá despreciar dicho término. Por lo tanto, en estos casos el resultado usando cualquiera de los métodos será similar pudiendo incluso llegar a una convergencia cuadrática para los problemas consistentes (aquéllos en los que $r(x^*) = 0$).

Por norma general la velocidad de convergencia local será mucho mayor en los métodos de tipo Newton que en los de Gauss-Newton. Sin embargo, es frecuente que el coste computacional que conlleva calcular las derivadas segundas en problemas de dimensión alta sea tan grande que sea imposible su utilización, teniendo que usar Gauss-Newton.

2.4. Métodos del tipo Gauss-Newton

2.4.1. El método de Gauss-Newton no amortiguado

El primer método que se va a analizar es el método de Gauss-Newton no amortiguado. La base es una aproximación lineal del vector residual (2.26). La sucesión $\{x_k\}_k$ de aproximaciones cada vez mejores se irá construyendo según

$$x_{k+1} = x_k + p_k, \tag{2.31}$$

donde p_k se elige como

$$\min_{p \in \mathbb{R}^n} \|r(x_k) + J(x_k)p\|_2. \tag{2.32}$$

Este problema (2.32) ya somos capaces de resolverlo, pues es de la forma (1.16) donde al igualar el gradiente a cero se llega a las ecuaciones normales

$$J^T(x_k)J(x_k)p_k = -J^T(x_k)r(x_k), \tag{2.33}$$

que podremos resolver sin construirlas aplicando por ejemplo la factorización QR a la matriz $J(x_k)$.

La gran ventaja de este método es que converge localmente muy rápido en problemas que sean casi lineales y consistentes. (Obviamente los problemas que sean lineales con este método se resuelven con una sola iteración.) Sin embargo, puede ser incapaz de resolver problemas que disten mucho de ser lineales o que tengan residuos grandes. Esto se puede observar en el siguiente ejemplo propuesto en [1].

Ejemplo 2.8. Consideramos el problema dado por

$$\begin{cases} r_1(x) = x + 1, \\ r_2(x) = \lambda x^2 + x - 1, \end{cases}$$

donde λ es un parámetro fijo del problema. Es claro que el mínimo de la suma de los cuadrados, $r_1^2(x) + r_2^2(x)$, se alcanza en $x^* = 0$. Vamos a ver que al aplicar el método de Gauss-Newton se llega a

$$x_{k+1} = \lambda x_k + O(x_k^2).$$

En primer lugar, sabemos que para este problema $m = 2$ y $n = 1$. Expresamos el vector residual de forma matricial y calculamos su derivada

$$r(x) = \begin{pmatrix} x + 1 \\ \lambda x^2 + x - 1 \end{pmatrix}, \qquad J(x) = \begin{pmatrix} 1 \\ 2\lambda x + 1 \end{pmatrix}.$$

Según (2.32), el método busca el vector p que cumpla

$$\min_p \left\| \begin{pmatrix} x + 1 \\ \lambda x^2 + x - 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 2\lambda x + 1 \end{pmatrix} p \right\|_2.$$

Resolviendo las ecuaciones normales (2.33) vamos a ser capaces de deducir la dirección de búsqueda p_k

$$\begin{pmatrix} 1 & 2\lambda x + 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2\lambda x + 1 \end{pmatrix} p_k = - \begin{pmatrix} 1 & 2\lambda x + 1 \end{pmatrix} \begin{pmatrix} x + 1 \\ \lambda x^2 + x - 1 \end{pmatrix}$$

$$\begin{aligned} p_k &= \frac{-(x_k + 1) - (2\lambda x_k + 1)(\lambda x_k^2 + x_k - 1)}{1 + (2\lambda x_k + 1)^2} \\ &= \frac{-2\lambda^2 x_k^3 - 3\lambda x_k^2 + 2(\lambda - 1)x_k}{1 + (2\lambda x_k + 1)^2}. \end{aligned}$$

Como lo que buscamos es probar $x_{k+1} = \lambda x_k + O(x_k^2)$ y según el método la nueva aproximación cumple $x_{k+1} = x_k + p_k$ bastaría con probar

$$p_k = (\lambda - 1)x_k + O(x_k^2).$$

Para ello veremos si es cierto que

$$p_k = \frac{-2\lambda^2 x_k^3 - 3\lambda x_k^2 + 2(\lambda - 1)x_k}{1 + (2\lambda x_k + 1)^2} = (\lambda - 1)x_k + O(x_k^2).$$

Multiplicando por el denominador y despejando obtenemos

$$-2\lambda^2 x_k^3 - 3\lambda x_k^2 + 2(\lambda - 1)x_k - (\lambda - 1)x_k(1 + (2\lambda x_k + 1)^2) = O(x_k^2),$$

luego se cumple. La dirección de búsqueda es

$$p_k = (\lambda - 1)x_k + O(x_k^2)$$

y por consiguiente

$$\begin{aligned} x_{k+1} &= x_k + p_k = x_k + (\lambda - 1)x_k + O(x_k^2) \\ &= \lambda x_k + O(x_k^2). \end{aligned}$$ □

Si analizamos la convergencia local del método de Gauss-Newton en el ejemplo anterior, nos damos cuenta de que no convergerá si $|\lambda| > 1$ pues se genera una sucesión de iterantes de módulo creciente. Esto nos lleva a buscar métodos mejores para tratar estos casos.

2.4.2. El método de Gauss-Newton amortiguado

Una mejora del anterior método es el denominado método de Gauss-Newton amortiguado. En éste las iteraciones se irán formado siguiendo

$$x_{k+1} = x_k + \alpha_k p_k, \tag{2.34}$$

donde p_k es la solución de (2.32) y α_k es la longitud de paso que debe ser determinada.

Observación 2.9. Si nos fijamos en la Sección 2.2 nos damos cuenta que estamos aplicando la estrategia de búsqueda de línea donde p_k es la dirección de búsqueda, ahora denominada dirección de Gauss-Newton, y α_k es la longitud de paso. Y es precisamente esta longitud de paso la mejora con respecto al método no amortiguado. De hecho, se podría considerar que este método engloba al anterior tomando $\alpha_k = 1$ para todo k. Por lo tanto, este método se basa en elegir adecuadamente p_k y α_k.

En primer lugar, veamos una manera de elegir p_k, incluso cuando el rango de la matriz $J(x_k)$ sea deficiente. Haciendo uso de la matriz pseudoinversa podemos reescribir (2.33) como

$$p_k = -J^+(x_k)r(x_k). \tag{2.35}$$

Con esta definición de la dirección de Gauss-Newton, podemos verificar que p_k cumple la siguiente propiedad.

Proposición 2.10. *Si x_k no es un punto crítico entonces p_k es una **dirección de descenso**, es decir, se verifica que $\|r(x_k + \alpha p_k)\|_2^2 < \|r(x_k)\|_2^2$ para un valor de α positivo suficientemente pequeño.*

Demostración. Partimos de la aproximación lineal

$$r(x_k + \alpha p_k) = r(x_k) + J(x_k)\alpha p_k + O(|\alpha|^2).$$

Usando que $p_k = -J^+(x_k)r(x_k)$ y la proyección ortogonal sobre el espacio columna de la matriz jacobiana $P_{J_k} = J(x_k)J(x_k)^+$, obtenemos

$$r(x_k + \alpha p_k) = r(x_k) - \alpha P_{J_k} r(x_k) + O(|\alpha|^2).$$

Al calcular su norma euclídea al cuadrado conseguimos

$$\|r(x_k + \alpha p_k)\|_2^2 = r(x_k)^T r(x_k) - 2\alpha r(x_k)^T P_{J_k} r(x_k) + O(|\alpha|^2),$$

donde se han utilizado las condiciones de Penrose 2 y 3 que satisface la pseudoinversa de una matriz (Ver Apéndice A: Teorema A.1).
Además, como

$$\begin{aligned}
\|P_{J_k} r(x_k)\|_2^2 &= r(x_k)^T P_{J_k}^T P_{J_k} r(x_k) = \\
&= r(x_k)^T (J(x_k)J^+(x_k))^T (J(x_k)J^+(x_k)) r(x_k) = \\
&= r(x_k)^T J(x_k)J^+(x_k)(J(x_k)J^+(x_k)) r(x_k) = \\
&= r(x_k)^T J(x_k)J^+(x_k) r(x_k) = r(x_k)^T P_{J_k} r(x_k)
\end{aligned}$$

entonces

$$\|r(x_k + \alpha p_k)\|_2^2 = \|r(x_k)\|_2^2 - 2\alpha \|P_{J_k} r(x_k)\|_2^2 + O(|\alpha|^2).$$

Por consiguiente, si x_k no es un punto crítico entonces $J(x_k)^T r(x_k) \neq 0$ y, usando la descomposición en valores singulares de $J(x_k)$, se llega a que $P_{J_k} r(x_k) \neq 0$. Con esto queda probado que p_k es una dirección de descenso para r.

Aclaremos la última implicación ($J(x_k)^T r(x_k) \neq 0 \Rightarrow P_{J_k} r(x_k) \neq 0$) demostrando que si $P_{J_k} r(x_k) = 0$ entonces $J(x_k)^T r(x_k) = 0$. Para ello, utilizaremos la notación de la Sección 1.3.5. Tenemos

$$J = U\Sigma V^T = \sum_{i=1}^{n} \sigma_i u_i v_i^T, \qquad J^T = V\Sigma^T U = \sum_{i=1}^{n} \sigma_i v_i u_i^T,$$

$$P_J = JJ^+ = U\Sigma\Sigma^+ U^T = \sum_{i=1}^{n} u_i u_i^T.$$

Luego P_J es la matriz de proyección sobre el subespacio generado por las primeras n columnas de U. Si $P_J\, r = 0$ entonces $\sum_{i=1}^{n}(u_i^T r)u_i = 0$ lo que implica que $u_i^T r = 0$ con $1 \leq i \leq n$. Es decir, r va a ser ortogonal a las primeras n columnas de U y por lo tanto $J^T r = \sum_{i=1}^{n} \sigma_i v_i (u_i^T r) = 0$. □

Por otra parte, este método necesita elegir adecuadamente la longitud de paso α_k. Como vimos en la Sección 2.2.3 existen distintas condiciones que nos ayudan a elegir unos adecuados α_k. Un algoritmo que se basa en estas condiciones es el de Armijo-Goldstein que enunciaremos a continuación, aunque se encuentra mucho más detallado en [12] p.491 y en [6] pp.100-102.

Algoritmo de Armijo-Goldstein
Para elegir la longitud de paso α_k del método amortiguado de Gauss-Newton se irá disminuyendo su valor siguiendo la serie geométrica de razón $1/q$ con $q > 1$ (generalmente se toma $q = 2$) hasta que se verifique la siguiente desigualdad

$$\|r(x_k)\|_2^2 - \|r(x_k + \alpha_k p_k)\|_2^2 \geq \frac{1}{2}\alpha \|J(x_k)p_k\|_2^2 \tag{2.36}$$

Nota 2.11. Este algoritmo se basa en las condiciones de Goldstein (2.25) y en las igualdades

$$\|J(x_k)p_k\|_2^2 = \|P_{J_k} r(x_k)\|_2^2 = r^T (JJ^+)^T JJ^+ r = \nabla f^T(x_k)p_k,$$

conseguidas aplicando algunas propiedades de las matrices pseudoinversas (Teorema A.2 del Apéndice A).

Otra forma de elegir la longitud α_k es buscar la solución del problema unidimensional

$$\min_{\alpha} \|r(x_k + \alpha p_k)\|_2^2. \tag{2.37}$$

Sin embargo, la solución de (2.37) no se puede conseguir en un número finito de pasos por lo que generalmente se considera una aproximación. Un ejemplo de posible aproximación ha sido desarrollado por Lindström y Wedin en [8] p.269. Ésta consiste en buscar la función $p(\alpha)$, aproximación de $r(x_k + \alpha p_k)$, determinada por

$$p(0) = f(0), \quad \nabla p(0) = \nabla f(0), \quad p(\alpha_0) = f(\alpha_0),$$

donde α_0 sea un valor inicial que se le da a la longitud de paso.
Una vez determinada esta aproximación $p(\alpha)$ se minimiza, es decir, se elige α_k como la solución de

$$\min_{\alpha} \|p(\alpha)\|_2. \tag{2.38}$$

De la misma forma que es importante hallar adecuadamente p_k y α_k para implementar el método, es fundamental determinar correctamente el rango de $J(x_k)$ a la hora de resolver (2.35) cuando el rango no es máximo. El siguiente ejemplo ilustra claramente esta importancia (Ver [6] p.136)

Ejemplo 2.12. Sean

$$J(x_k) = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \quad \text{y} \quad r(x_k) = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

con $\varepsilon \ll 1$ y r_1, r_2 de tamaño unidad ($O(1)$).

Si consideramos que la matriz $J(x_k)$ es de rango dos entonces la dirección de búsqueda será

$$p_k = -\begin{pmatrix} r_1 \\ r_2/\varepsilon \end{pmatrix}, \text{ pues } J^+ = \begin{pmatrix} 1 & 0 \\ 0 & 1/\varepsilon \end{pmatrix}. \tag{2.39}$$

Sin embargo, si se considera que tiene rango uno entonces

$$p_k = -\begin{pmatrix} r_1 \\ 0 \end{pmatrix}, \text{ pues } J^+ = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.40}$$

Las direcciones obtenidas al variar el rango en una unidad son ortogonales (si tenemos en cuenta que r_1 es despreciable frente a r_2/ε), lo que muestra la importancia de determinar correctamente el rango. De hecho, la primera dirección es casi ortogonal al gradiente, $J^T(x_k)r(x_k)$, lo que provocaría una convergencia muy lenta del método si se utilizara.

✣ Convergencia

En la mayoría de los problema el método amortiguado de Gauss-Newton es localmente convergente. De hecho, suele alcanzar el mínimo global. Sin embargo, la velocidad de convergencia puede llegar a ser muy lenta en problemas con grandes residuos o que disten mucho de ser lineales.

Empezemos analizando la convergencia local. El método no amortiguado de Gauss-Newton se puede ver como una iteración de punto fijo

$$x_{k+1} = F(x_k), \quad F(x) = x - J(x)^+ r(x), \tag{2.41}$$

donde se ha usado (2.35).

El gradiente de la función $F(x)$ evaluado en el mínimo x^*, usando la notación de la Sección 1.1, es

$$\nabla F(x^*) = -[J(x^*)^T J(x^*)]^{-1} Q(x^*) \tag{2.42}$$

Lo anterior se deduce desarrollando (2.41) como

$$F(x) = x - [J(x)^T J(x)]^{-1} J(x)^T r(x) = x - [J(x)^T J(x)]^{-1} \nabla f(x) \tag{2.43}$$

y derivando y evaluando en x^*, pues el término que vendría de derivar la matriz $[J(x)^T J(x)]^{-1}$ vendría multiplicado por $\nabla f(x^*) = 0$. Se obtiene, por tanto

$$\nabla F(x^*) = I - [J(x^*)^T J(x^*)]^{-1} \nabla^2 f(x^*). \tag{2.44}$$

Utilizando ahora (1.8) se obtiene

$$\begin{aligned} \nabla F(x^*) &= I - [J(x^*)^T J(x^*)]^{-1} [J(x^*)^T J(x^*) + Q(x^*)] \\ &= -[J(x^*)^T J(x^*)]^{-1} Q(x^*). \end{aligned} \tag{2.45}$$

La velocidad de convergencia, siguiendo los criterios de una iteración de punto fijo, sabemos que va a depender del radio espectral de la matriz $\nabla F(x^*)$. Recordamos que el radio espectral es el máximo de los valores absolutos de los autovalores de la matriz. Además, como veremos a continuación, $\nabla F(x^*)$ tiene los mismos autovalores que $(J(x^*)^+)^T Q(x^*) J(x^*)^+ = \gamma K(x^*)$, donde K es la matriz de curvatura (2.4). Por lo tanto, se puede decir que la velocidad de convergencia estará acotada por

$$\rho = \rho(\nabla F(x^*)) = \gamma \, \text{máx}(\sigma_1, -\sigma_n), \tag{2.46}$$

donde $\gamma = \|r(x^*)\|_2$ y σ_i son los inversos de los autovalores no nulos de K.

Para ver que las matrices $\nabla F(x^*)$ y $(J(x^*)^+)^T Q(x^*) J(x^*)^+$ tienen los mismos autovalores no nulos veamos que si v es autovector de $(J^T J)^{-1} Q$ con autovalor λ entonces Jv es autovector de $(J^+)^T Q J^+$ con autovector λ.

$$\begin{aligned} (J^+)^T Q J^+ (Jv) &= (J^+)^T Q (J^T J)^{-1} J^T (Jv) = (J^+)^T Q v = ((J^T J)^{-1} J^T)^T Q v \\ &= J (J^T J)^{-1} Q v = J \lambda v = \lambda (Jv). \end{aligned} \tag{2.47}$$

Como $\nabla F(x^*) = -[J(x^*)^T J(x^*)]^{-1} Q(x^*)$, visto en (2.45), se concluye que $\nabla F(x^*)$ tiene los mismos autovalores no nulos que $\gamma K(x^*)$.

Cabe señalar que $\nabla F(x^*)$ es una matriz $n \times n$ mientras que K es $m \times m$, pero ambas tienen el mismo rango, n, igual al número de autovalores no nulos de ambas matrices.

En resumen,

$$\frac{\|p_k\|_2}{\|p_{k-1}\|_2} = \frac{\|x_{k+1} - x_k\|_2}{\|x_k - x_{k-1}\|_2} \leq \rho + O(\|x_k - x_{k-1}\|_2^2). \tag{2.48}$$

De hecho, como se afirma en [1] p.346, el radio espectral de la matriz $\nabla F(x^*)$, responsable de la velocidad de convergencia de la iteración se puede estimar durante la iteración mediante el cociente

$$\frac{\|J(x_{k+1}) p_{k+1}\|_2}{\|J(x_k) p_k\|_2} \leq \rho + O(\|x_k - x^*\|_2^2). \tag{2.49}$$

Podemos concluir, en base a lo obtenido, que el método no amortiguado de Gauss-Newton en general convergerá linealmente, pero si $\gamma = \|r(x^*)\|_2 = 0$ entonces la convergencia será superlineal. Además, la convergencia será bastante rápida si la norma residual $\|r(x^*)\|_2$ es pequeña o si $r(x)$ no está lejos de ser lineal, es decir, si $\|G_i\|_2$ es pequeña para todo i. Sin embargo, si ρ es mayor que 0.5 la convergencia será lenta siendo aconsejable usar otro método que utilice la información que proporciona la segunda derivada o incluso cambiar el modelo inicialmente considerado, $g(x)$.

En el caso de que el método de Gauss-Newton tenga una línea de búsqueda exacta la velocidad de convergencia va a ser, según Ruhe (ver [14] p.361)

$$\hat{\rho} = \gamma(\sigma_1 - \sigma_n)/(2 - \gamma(\sigma_1 + \sigma_n)), \tag{2.50}$$

donde

$$\hat{\rho} = \begin{cases} = \rho & \text{si } \sigma_n = -\sigma_1, \\ < \rho & \text{en otro caso.} \end{cases} \tag{2.51}$$

Además, si $\gamma\sigma_1 < 1$ entonces $\hat{\rho} < 1$, lo que implica que siempre se conseguirá una convergencia cerca de un mínimo local. Esto supone una mejora con respecto al método no amortiguado.

2.4.3. El método de Levenberg-Marquardt

Como se ha dicho anteriormente, existen casos en los que el método amortiguado de Gauss-Newton, cuando la matriz jacobiana no tiene rango columna completo, puede fallar. Una alternativa distinta, sin tener que evaluar las derivadas segundas del vector residual, sería darle más estabilidad al método amortiguado. El método que se basa en esta idea es el de Levenberg-Marquardt, el cual fue el precursor de los después denominados métodos de la región de confianza.

Como se desarrolla en [10], el algoritmo busca una aproximación a un modelo lineal. Partiendo de un punto dado $x_k \in \mathbb{R}^n$, nos gustaría minimizar

$$\Psi(p_k) = \|r(x_k + p_k)\|_2, \tag{2.52}$$

siendo entonces $x_k + p_k$ la solución deseada. Es aquí cuando, a pesar de que Ψ es no lineal, se busca linealizar $r(x_k + p_k)$ llegando a un problema lineal de mínimos cuadrados de la forma

$$m_k(p_k) = \|r(x_k) + J(x_k)p_k\|_2. \tag{2.53}$$

Está claro que la linealización no va a ser válida para cualquier valor de p_k, por lo tanto debemos añadir la condición

$$\|D_k p_k\|_2 \leq \delta_k, \tag{2.54}$$

donde δ_k marca el radio de la región de confianza como vimos en (2.10) y D_k es una matriz no singular que tiene en cuenta el problema del escalado.

Observación 2.13. Como bien se explica en [11] p.27, a la hora de ejecutar un algoritmo es muy importante tener en cuenta el denominado problema del escalado. Un problema se dice que no está bien escalado si al cambiar el vector x en una determina dirección se produce un cambio mayor que si se hubiera realizado el mismo cambio en una dirección distinta del mismo vector. Para tener en cuenta estas distintas velocidades de cambio de las componentes del vector residual se suele añadir una matriz no singular, generalmente diagonal con valores positivos en la diagonal, denotada por D_k que proporciona una invarianza de escala. Estaríamos pasando de una región de confianza circular a una elipsoidal donde, si D_k es diagonal, la longitud de los ejes principales del elipsoide coincidirán con las direcciones de coordenadas.

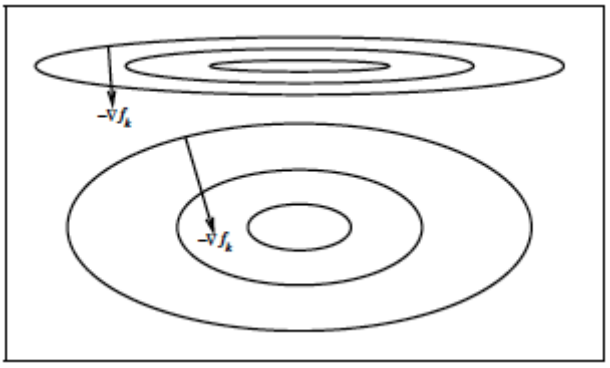

Figura 2.4: Representación de dos problema, uno con problema de escala (superior) y otro con la corrección de escala (inferior).

La elección de δ_k dependerá del radio entre la reducción actual y la predicha, es decir, de

$$\rho_k = \frac{\|r(x_k)\|_2^2 - \|r(x_k + p_k)\|_2^2}{\|r(x_k)\|_2^2 - \|r(x_k) + J(x_k)p_k\|_2^2}, \tag{2.55}$$

que mide el parecido entre el modelo lineal m_k y la función no lineal Φ.

A continuación mostramos una posible implementación del algoritmo de Levenberg-Marquardt desarrollada en [10] pp.8-9 y [11] p.69.

Algoritmo 2.14. **Algoritmo de Levenberg-Marquardt:**

Dados unos valores x_0, D_0 y δ_0 iniciales y un valor $\beta \in (0, 1)$, para $k = 1, 2, \ldots$

1. Se calcula $\|r(x_k)\|_2^2$.
2. Se determina la solución p_k de

$$\min_p \|r(x_k) + J(x_k)p\|_2^2, \text{ con } \|D_k p\|_2 \leq \delta_k, \tag{2.56}$$

donde D_k es una matriz diagonal de escala.

3. Se calcula el valor de ρ_k dado por (2.55).

4. Si $\rho_k \leq \beta$ entonces $x_{k+1} = x_k$ y $J_{k+1} = J_k$.
   Si $\rho_k > \beta$ entonces $x_{k+1} = x_k + p_k$ y se calcula J_{k+1}.

5. Se actualizan D_k y δ_k.

   Si $\rho_k \leq 1/4$ entonces se escoge $\delta_{k+1} \in [\frac{1}{10}\delta_k, \frac{1}{2}\delta_k]$.
   Si $\rho_k \in (\frac{1}{4}, \frac{3}{4})$ y $\lambda = 0$, o si $\rho_k > 3/4$ entonces $\delta_{k+1} = 2\|D_k p_k\|_2$.
   En otro caso, $\delta_{k+1} = \delta_k$.

   $D_{k+1} = diag(d_1^{(k+1)}, \ldots, d_n^{(k+1)})$, donde $d_i^{(k+1)} = \text{máx}\,\{d_i^{(k)}, \|\partial_i r(x_{k+1})\|_2\}$.

Esto quiere decir, que cambiaremos las dimensiones de la región de confianza si se alcanza su frontera en la iteración anterior (aumentando las dimensiones) o si se ha perdido la aproximación lineal deseada (en cuyo caso se reducirán).

Para resolver el paso 2 en el Algoritmo 2.14, según se explica en [11] pp. 258-261, se calcula la solución del problema de Gauss-Newton

$$\min_p \|r(x_k) + J(x_k)p\|_2^2 \tag{2.57}$$

y si la solución ya está dentro de la región de confianza ya habríamos acabado, siendo ésta la solución de nuestro problema. Sin embargo, si la solución de (2.57) se encuentra fuera de la región de confianza, se puede probar que existe un valor $\lambda > 0$ tal que la solución para (2.56) cumple

$$(J(x_k)^T J(x_k) + \lambda D_k^2)p_k = -J(x_k)^T r(x_k), \tag{2.58}$$

o lo que es lo mismo, es solución de

$$\min_p \left\| \begin{bmatrix} J(x_k) \\ \sqrt{\lambda} D_k \end{bmatrix} p + \begin{bmatrix} r(x_k) \\ 0 \end{bmatrix} \right\|^2. \tag{2.59}$$

Un algoritmo para resolver (2.58) se encuentra detallado en [11] p.259.

✣ Convergencia

Sobre la convergencia local, este método se comportará de forma similar al de Gauss-Newton. En ambos se ha utilizado la aproximación de la matriz hessiana $\nabla^2 f(x) = J(x)^T J(x)$ despreciando el segundo término en (1.8).

Según Moré es cierto el siguiente resultado que se encuentra demostrado en el Capítulo 4 de [11].

Teorema 2.15. *Sea $r(x)$ continuamente diferenciable en $\mathbb{R}^n$, $J(x)$ uniformemente continua y $J(x_k)$ acotada. Entonces el algoritmo de Levenberg-Marquardt 2.14 converge hacia un punto crítico.*

Versiones mejoradas del método, como las realizadas por Powell y Osborne, han conseguido lograr incluso una convergencia global.

A pesar de estos resultados sobre la convergencia, en problemas con residuos grandes o no muy lineales la convergencia todavía es muy lenta siendo necesario en muchos casos utilizar métodos más robustos y complejos que, usando las segundas derivadas, consigan una convergencia más rápida.

2.5. Métodos del tipo Newton

Como se introdujo en la Sección 2.3, existen métodos más robustos y complejos que utilizando las derivadas segundas permiten resolver ciertos casos cuya aproximación era muy lenta o que no se podían resolver mediante Gauss-Newton.

El método de Newton se basa en utilizar el punto crítico del modelo cuadrático (2.28) de $f(x)$ evaluado en la iteración actual, para definir la siguiente aproximación. Es decir, el método de Newton se puede ver como una búsqueda de línea, $x_{k+1} = x_k + \alpha_k p_k$, donde la dirección de búsqueda p_k queda determinada por (2.16) y la longitud de paso α_k se toma igual a 1.

Por lo tanto, desarrollando $\nabla^2 f(x) = J(x_k)^T J(x_k) + Q(x_k)$ será necesario resolver

$$(J(x_k)^T J(x_k) + Q(x_k))p_k = -J(x_k)^T r(x_k), \tag{2.60}$$

para así poder obtener p_k.

✣ Convergencia

Respecto a la convergencia local del método de Newton, como demuestran Dennis y Schnabel en [4] pp.90 y 229, es cierto el siguiente resultado.

Teorema 2.16. *Si $\nabla^2 f(x)$ es lipschitziana en un entorno abierto de x^* y $\nabla^2 f(x^*)$ es definida positiva entonces el método de Newton tendrá convergencia local cuadrática.*

La convergencia global se podría conseguir en algunos casos siguiendo la búsqueda de línea explicada anteriormente con p_k determinado por (2.60).

Observación 2.17. La matriz $J(x_k)^T J(x_k) + Q(x_k)$ debe ser definida positiva para garantizar que la dirección de p_k sea de descenso.

Así pues, las propiedades de convergencia local de este método son bastante mejores que las de los métodos anteriores, puesto que el método de Newton es rápidamente localmente convergente en casi todos los problemas, mientras que el amortiguado de Gauss-Newton o los métodos de Levenberg-Marquardt pueden ser lentamente localmente convergentes si $r(x)$ es poco lineal o $r(x^*)$ es grande. Sin embargo, el método de Newton se usa pocas veces en problemas de mínimos cuadrados no lineales por lo costoso que es el cálculo de las derivadas segundas, las cuales muchas veces no son analíticamente accesibles. Supongamos que el cálculo de la matriz jacobiana $J(x)$ no es analíticamente viable, entonces se tendría que usar una aproximación por diferencias finitas, lo que supondría un coste adicional de n evaluaciones de $r(x)$ por iteración. En estos casos, si luego necesitásemos calcular la hessiana $\nabla^2 f(x)$ para aplicar el método de Newton los costes se harían enormes, teniendo que hacer $(n^2 + 3n)/2$ evaluaciones de $r(x)$ a mayores en cada iteración. Por este motivo, generalmente se usan métodos mejores que se explicarán a continuación.

Las características principales que se obtienen con este método se pueden resumir en:

- ✯ Ventajas:
  - Convergencia cuadrática en el caso de que $J(x)$ sea no singular y se tome un buen iterante inicial x_0.
- ✯ Desventajas:
  - Para muchos problemas no se consigue la convergencia global deseada.
  - Se necesita calcular $J(x_k)$ y $Q(x_k)$ en cada iteración, lo que es muy costoso.
  - En cada iteración hay que resolver un sistema de ecuaciones lineales, (2.60), que puede ser singular o estar mal condicionado.

2.5.1. El método de Newton híbrido

Una mejora del método anterior es el denominado método de Newton híbrido. Este, como explican Gill y Murray en [1] p.348, es menos costoso que el anterior pues en vez de calcular directamente la matriz hessiana como $\nabla^2 f(x_k) = J(x_k)^T J(x_k) + Q(x_k)$ utilizan la aproximación dada por $J(x_k)^T J(x_k)$ en el subespacio de autovalores grandes de $J(x_k)$.

La implementación de este método parte de la descomposición en valores singulares de la matriz jacobiana

$$J(x_k) = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T, \quad U \in \mathbb{R}^{m \times m}, \quad V \in \mathbb{R}^{n \times n}, \tag{2.61}$$

donde $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$ con $\sigma_1 \geq \ldots \geq \sigma_n$.

Usando la misma ecuación que el método de Newton para encontrar la dirección de búsqueda (2.60) y premultiplicando por V^T llegamos a

$$\begin{aligned} (J(x_k)^T J(x_k) + Q(x_k))p_k &= -J(x_k)^T r(x_k) \\ \Rightarrow V^T(V\Sigma U^T U \Sigma V^T + Q(x_k))Vq_k &= -V^T(V\Sigma U^T)r(x_k) \\ \Rightarrow (\Sigma^2 + V^T Q(x_k)V)q_k &= -\Sigma\varsigma \end{aligned} \tag{2.62}$$

donde hemos denotado por ς a las primeras n componentes del vector $U^T r(x_k)$ y q_k al vector que verifica $p_k = Vq_k$.

Ahora si separamos en dos partes a los valores singulares, en grandes y pequeños, denotándolos como

$$\Sigma = diag(\Sigma_1, \Sigma_2), \quad \text{con} \quad \begin{aligned} \Sigma_1 &= diag(\sigma_1, \ldots, \sigma_p) \\ \Sigma_2 &= diag(\sigma_{p+1}, \ldots, \sigma_n) \end{aligned}$$

y de la misma forma a V, q_k y ς, entonces las p primeras ecuaciones de (2.62) se pueden escribir como

$$(\Sigma_1^2 + V_1^T Q(x_k)V_1)q_1 + V_1^T Q(x_k)V_2 q_2 = -\Sigma_1\varsigma_1. \tag{2.63}$$

Si, como hemos dicho anteriormente, despreciamos $Q(x_k)$ en el subespacio correspondiente a los valores singulares más grandes de $J(x_k)$ entonces podemos resolver lo anterior obteniendo

$$q_1 = -\Sigma_1^{-1}\varsigma_1. \tag{2.64}$$

Sustituyendo lo obtenido en las últimas $n-p$ ecuaciones de (2.62) podemos deducir el valor de q_2 resolviendo

$$(\Sigma_2^2 + V_2^T Q(x_k)V_2)q_2 = -\Sigma_2\varsigma_2 - V_2^T Q(x_k)V_1 q_1. \tag{2.65}$$

Por lo tanto, la aproximación a la dirección de Newton que se encuentra usando este método es

$$p_k = Vq_k = V_1 q_1 + V_2 q_2 \tag{2.66}$$

donde hemos visto que q_1 se deduce de (2.64) y q_2 de (2.65).

Cabe señalar que la división en dos de los valores singulares se actualiza en cada iteración buscando siempre incluir el mayor número posible de valores singulares dentro de la submatriz Σ_1, considerándolos valores singulares grandes, siempre y cuando se pueda conseguir con ellos un progreso adecuado en la búsqueda del mínimo.

Además de esta alternativa al método de Newton existen muchas otras como por ejemplo utilizar una aproximación de Quasi-Newton de la matriz $Q(x_k)$. La forma en la que se puede construir esta aproximación es semejante al método explicado a continuación.

2.5.2. El método de Quasi-Newton

Las rutinas de optimización en este método se basan en usar aproximaciones de la matriz hessiana a base de realizar varias evaluaciones sucesivas del gradiente de la función, consiguiendo en numerosas ocasiones una convergencia superlineal. Para encontrar una buena aproximación simétrica, S_k, de la matriz hessiana en la iteración k-ésima será necesario que esta matriz se aproxime bastante a la curvatura que toma f en el paso de x_{k-1} a x_k. Por ello, se toma S_k como la matriz que verifica

$$S_k(x_k - x_{k-1}) = J(x_k)^T r(x_k) - J(x_{k-1})^T r(x_{k-1}). \tag{2.67}$$

La ecuación anterior se denomina **relación de Quasi-Newton**.

Con esto, la dirección de búsqueda se obtiene resolviendo

$$S_k p_k = -J(x_k)^T r(x_k), \tag{2.68}$$

donde se ha cogido la fórmula de la dirección de Newton (2.16) y se ha reemplazado la matriz hessiana $\nabla^2 f(x_k)$ por S_k.

Nota 2.18. Como aproximación inicial es recomendable tomar $S_0 = J(x_0)^T J(x_0)$. Además, el método trata de partiendo de la aproximación hecha en una iteración construir la siguiente, adecuando la curvatura de manera que se destruya la menor cantidad posible de información almacenada en la aproximación anterior. Es decir, se trata de actualizar la aproximacion S_k de modo que la diferencia entre S_{k-1} y S_k sea una matriz de rango pequeño.

2.6. Elección del método

Uno de los mayores interrogantes que surgen al ver los métodos anteriores consiste en saber que método es el mejor para conseguir una convergencia global y rápida al enfrentarnos a un determinado problema. Ramsin y Wedin en [1] p.350 dan ciertas recomendaciones para saber elegir entre los métodos de Gauss-Newton y Quasi-Newton. Estas recomendaciones son las bases para los métodos híbridos que permiten cambiar de un método a otro según convenga.
La regla se basa en analizar el radio espectral ρ para el método de Gauss-Newton usando (2.49).

- Para problemas sencillos

  1. Si $\rho \leq 0.5$ entonces es mejor utilizar el método de Gauss-Newton.
  2. Si $\rho > 0.5$ entonces el método de Quasi-Newton propociona una mejor convergencia.

- Para problemas complejos
  1. Si $\rho \leq 0.7$ entonces el método de Gauss-Newton es más rápido.
  2. Si $\rho > 0.7$ entonces el método de Quasi-Newton asegura una mejor convergencia.

A partir de este criterio surgen distintas mejoras de los métodos anteriores.

Mejoras

En la práctica, la aproximación Quasi-Newton no es muy eficiente pues menosprecia la información que nos proporciona la matriz jacobiana, $J(x_k)$, la cual en numerosas ocasiones es el factor dominante en el cálculo de la matriz hessiana mediante el término $J(x_k)^T J(x_k)$. Por este motivo, Dennis, Gay y Welsch en [3] realizan una mejora definiendo la matriz S_k como

$$S_k = J(x_k)^T J(x_k) + B_k, \tag{2.69}$$

donde B_k verifica

$$B_k(x_k - x_{k-1}) = z_k \quad \text{con} \quad z_k = J(x_k)^T r(x_k) - J(x_{k-1})^T r(x_k). \tag{2.70}$$

La solución de esta ecuación que minimiza los cambios con respecto a B_{k-1}, como explican Dennis y Schnabel en [4] pp.231-232, viene dada por

$$B_k = B_{k-1} + \frac{(z_k - B_{k-1}\Delta x)y_k^T + y_k(z_k - B_{k-1}\Delta x)^T}{y_k^T \Delta x} - \frac{(z_k - B_{k-1}\Delta x)^T \Delta x \; y_k y_k^T}{(y_k^T \Delta x)^2}, \tag{2.71}$$

donde $\Delta x = x_k - x_{k-1}$ e $y_k = J(x_k)^T r(x_k) - J(x_{k-1})^T r(x_{k-1})$.

Esta actualización de la matriz B_k es utilizada en la subrutina denomina NL2SOL. Además, según el teorema siguiente que puede encontrarse en [4] p.206, se puede conseguir, siguiendo este método, una convergencia superlineal.

Teorema 2.19. *Sea f una función de clase $\mathscr{C}^2$ en un conjunto abierto convexo $D \subset \mathbb{R}^n$ y sea $\nabla^2 f$ lipschitziana en dicho conjunto. Supongamos que existe x^* tal que $\nabla f(x^*) = 0$ y $\nabla^2 f(x^*)$ es regular y definida positiva siendo S_k una aproximación simétrica de esta matriz hessiana. Entonces existirán $\varepsilon, \delta \geq 0$ tales que si $\|x_0 - x^*\|_2 < \varepsilon$ y $\|S_k - \nabla^2 f(x^*)\|_2 \leq \delta$ entonces $\{x_k\}_k$, dada por (2.71), está bien definida en D y converge superlinealmente a x^*.*

Observación 2.20. Un fallo que tiene la actualización (2.71) es que no tiene en cuenta los casos en que en el proceso de búsqueda del mínimo se lleguen a tener residuos muy cercanos a cero, es decir, los problemas de escala. Para evitar este inconveniente se puede reemplazar la matriz B_{k-1} por $\tau_k B_{k-1}$ en cada iteración (2.71) donde

$$\tau_k = \text{mín}(1, \frac{|\Delta x^T y_k|}{|\Delta x^T B_{k-1} \Delta x|}). \tag{2.72}$$

El código NL2SOL, anteriormente citado, combina los métodos de Gauss-Newton y de Quasi-Newton, con la mejora (2.71), decidiendo para cada iteración que método es mejor. Para ello, calcula las reducciones que proporcionan los dos modelos, comparándolas con la redución verdadera $f(x_{k+1}) - f(x_k)$, y elige la que más se aproxime. Generalmente, para las primeras iteraciones utiliza el método de Gauss-Newton hasta que la información que nos proporciona S_k se hace significativa cambiando al método de Quasi-Newton. Además, este código implementa la estrategia de región de confianza para asegurarse una convergencia global.

Otra alternativa en la que se usa la información que nos proporciona la segunda derivada es la de Ruhe, [14] p.362, donde usando el método de gradiente conjugado logra alcanzar una convergencia cuadrática, mucho más rápida que el método de Gauss-Newton, en ciertos problemas complejos. Sin embargo, en problemas que posean residuos pequeños este método malgastaría mucho tiempo sin conseguir mejores resultados, siendo entonces más recomendable utilizar el método amortiguado de Gauss-Newton.

Capítulo 3

Una aplicación en Electrónica

En este capítulo se describirá un problema físico cuya solución da lugar a un problema no lineal de mínimos cuadrados que puede resolverse numéricamente mediante los métodos estudiados en el capítulo anterior. En este caso emplearemos el método de Levenberg-Marquardt para encontrar los valores de los ocho parámetros (variables de estado) desconocidos que aparecen en un modelo matemático propuesto por E. Miranda. Este modelo trata de describir el funcionamiento de un memristor y es actualmente el modelo más aceptado por la comunidad científica. La explicación más detallada del modelo se puede encontrar en [9]. Además, como disponemos de datos experimentales obtenidos en el laboratorio, con este procedimiento podremos comprobar la validez del modelo propuesto, ya que en el método de mínimos cuadrados el objetivo principal consiste en reducir lo máximo posible las discrepancias con los datos experimentales. Incluso de esta forma podremos obtener pistas sobre qué caminos de investigación tomar para mejorar, si es posible, este modelo matemático preestablecido.

3.1. Un modelo matemático para memristores

En un primer lugar, describiremos brevemente qué es un memristor para luego explicar el modelo de E. Miranda con el que se trata de describir su funcionamiento.

3.1.1. Memristor

El memristor es una resistencia con memoria o un resistor capaz de variar el valor de su resistividad en función de la corriente eléctrica que circula a través de él y de la que ha circulado en el pasado. Con esta cualidad es capaz de almacenar información al mantener su valor de resistividad constante incluso cuando la corriente ha dejado de circular por él repentinamente, es decir, posee una memoria no volátil. Es considerado el cuarto elemento pasivo con el que se puede completar una serie de relaciones matemáticas entre las cuatro variables eléctricas

fundamentales: la corriente eléctrica (I), el voltaje (V), la carga eléctrica (q) y el flujo magnético (ϕ).

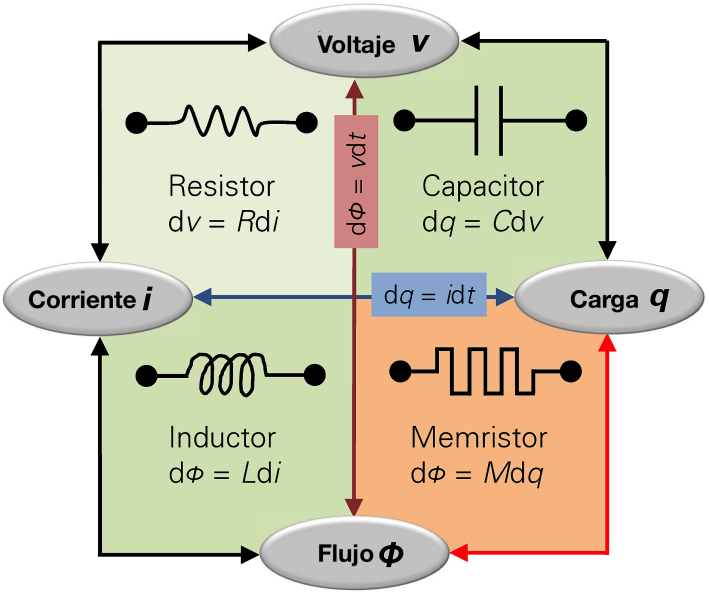

Figura 3.1: Los cuatro elementos pasivos básicos de un circuito eléctrico: resistencia, condensador, bobina y memristor.

Las ecuaciones genéricas con las que se puede describir cualquier dispositivo memristivo son

$$\begin{cases} y(t) = g(x,u,t)u(t) \\ \dfrac{dx}{dt} = f(x,u,t) \end{cases} \tag{3.1}$$

donde f y g son funciones continuas.
$u(t)$ es la señal de entrada (en nuestro caso será el voltaje).
$y(t)$ es la señal de salida (en nuestro caso será la corriente).
x es el conjunto de parámetros o variables de estado del modelo (en el modelo de Miranda hay 8 variables de estado).

3.1.2. El modelo de E. Miranda

Las ecuaciones fundamentales de este modelo son

$$I = sgn(V)[(\alpha R)^{-1} W\{\alpha R I_0(\lambda) e^{\alpha(|V|+RI_0(\lambda))}\} - I_0(\lambda)] \tag{3.2}$$

donde W es la función de Lambert,
sgn nos indica el signo,
α, R e I_0 son parámetros del modelo (variables de estado),

y la ecuación de estado

$$\frac{d\lambda}{dt} = g(\lambda)max(0,\dot{V}) + h(\lambda)min(0,\dot{V}) \tag{3.3}$$

donde λ es la variable de estado interno del memristor ($\lambda \in (0,1)$),
$\dot{V}$ es la derivada temporal de V,
g y h son funciones descritas en [9].

En este TFG no trataremos el origen y desarrollo de las ecuaciones de este modelo, lo cual se encuentra claramente explicado en [9] junto con la explicación del uso de determinadas aproximaciones. Aquí sólo expondremos las ecuaciones finales del modelo de Miranda necesarias para su posterior implementación en el programa de Matlab con el que se resolverá el problema no lineal de mínimos cuadrados que permitirá obtener los valores de las 8 variables de estado. Describimos a continuación los pasos necesarios, en el mismo orden que se puede observar a la hora de programar (Programa 1). Son los siguientes

❖ **Paso 1:**

La conducción por el memristor se produce a través de unos canales (filamentos conductores) que se van creando y destruyendo dependiendo del voltaje aplicado. Las curvas que describen estos procesos son

$$\Gamma^{\pm}(V) = [1 + e^{-\eta^{\pm}(V-V^{\pm})}]^{-1}, \tag{3.4}$$

donde los parámetros $\eta^{\pm}$ marcan la rápidez del cambio de estado, η^{+} en la formación (Γ^{+}) y η^{-} en la destrucción (Γ^{-}) de los canales.

❖ **Paso 2:**

Ahora calculamos la variable del estado interno del memristor denotada por λ. Para ello, se utiliza la siguiente fórmula recursiva

$$\begin{cases} \lambda_t = \text{mín}\{\Gamma^{-}[V_t], \text{máx}\{\lambda_0, \Gamma^{+}[V_t]\}\}, & t = 1,2,3,\dots \\ \lambda_0 = \lambda(V(t_0)), \end{cases} \tag{3.5}$$

donde λ_t y V_t son los valores discretizados de $\lambda(t)$ y $V(t)$ respectivamente, recogidos en un experimento de laboratorio. Es necesario incluir como condición inicial el estado interno del memristor a tiempo cero. Este valor lo consideraremos conocido a la hora de realizar un experimento.

❖ **Paso 3:**

E. Miranda considera que el memristor se puede entender como un diodo en serie con una resistencia, por lo que la amplitud de corriente que circularía por ese diodo se calcula, en función del estado interno del memristor, como

$$I_0(\lambda) = I_{0max}\lambda + I_{0min}(1-\lambda), \tag{3.6}$$

donde I_{0max} e I_{0min} son las corrientes máxima y mínima que circulan por el memristor.

❖ **Paso 4:**

Calculamos la corriente que según el modelo atravesaría al memristor

$$I = sgn(V)[(\alpha R)^{-1}W\{\alpha R I_0(\lambda)e^{\alpha(|V|+RI_0(\lambda))}\} - I_0(\lambda)], \tag{3.7}$$

donde usaremos la aproximación de Hermite-Padé para calcular la función de Lambert. Esta aproximación es

$$W(x) \approx ln(1+x)\left(1 - \frac{ln(1+ln(1+x))}{2+ln(1+x)}\right). \tag{3.8}$$

De las ecuaciones (3.4)-(3.8) se sigue que en el modelo de Miranda intervienen 8 parámetros que hay que determinar para poder calcular la corriente a partir del voltaje experimental. Estos 8 parámetros junto con la notación empleada en los distintos programas de Matlab del Apéndice B son los siguientes:

1. La velocidad de creación η^+ (denotada por `np`).
2. La velocidad de destrucción η^- (denotada por `nm`).
3. El voltaje medio de creación V^+ (denotado por `vp`).
4. El voltaje medio de destrucción V^- (denotado por `vm`).
5. El parámetro de determinación del mecanismo de conducción física específico del memristor utilizado α (denotado por `a`).
6. La resistencia del memristor R (denotada por `R`).
7. La corriente máxima que atraviesa al memristor (denotada por `imax`).
8. La corriente mínima que atraviesa al memristor (denotada por `imin`).

3.2. Procedimiento experimental

En primer lugar, es necesaria la recopilación de una gran cantidad de datos experimentales con los que poder analizar el modelo de Miranda anteriormente expuesto. El memristor utilizado posee una estructrua MIM (metal-aislante-metal) de TiN/Ti/HfO_2/W con un grosor de la capa de HfO_2 de 10 nm y un área total de 5x5 μm^2. Con este dispositivo se han realizado dos experimentos en los laboratorios del Departamento de Electricidad y Electrónica de la Universidad de Valladolid (UVA). En cada uno se aplicó una señal de entrada diferente, siendo en el primer experimento una sinusoidal con amplitud creciente (señal I) mientras que en el segundo fue una sinusoidal con amplitud decreciente (señal II). La duración de recogida de datos ha sido de 239.9 s en cada experimento en los que se ha podido medir en 2400 ocasiones el voltaje y la corriente que atravesaba al memristor, así como el tiempo en el que se tomaba cada dato. A continuación, se muestran las gráficas experimentales resultantes.

✣ **Primer experimento** :

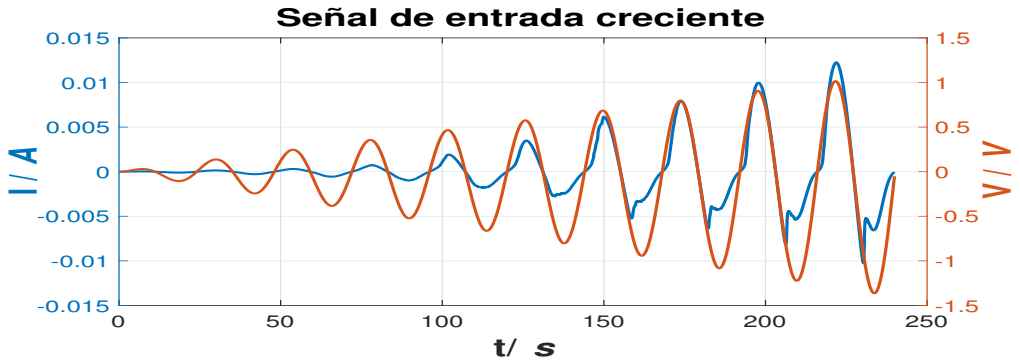

Figura 3.2: Variación del voltaje y la corriente en el memristor durante el transcurso del primer experimento. La señal de entrada es sinuosoidal creciente.

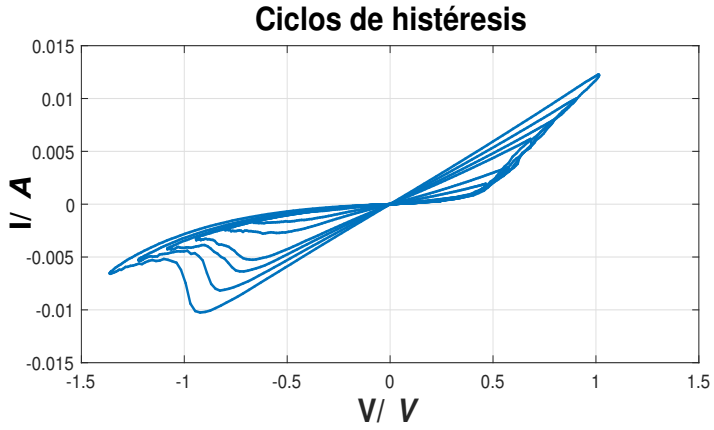

Figura 3.3: Relación Voltaje-Corriente experimental al aplicar la señal I.

✣ **Segundo experimento** :

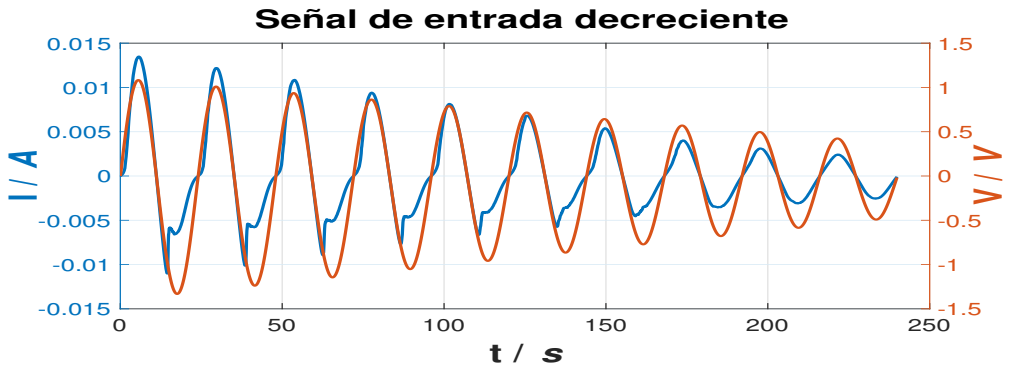

Figura 3.4: Variación del voltaje y la corriente en el memristor durante el transcurso del segundo experimento. La señal de entrada es sinuosoidal decreciente.

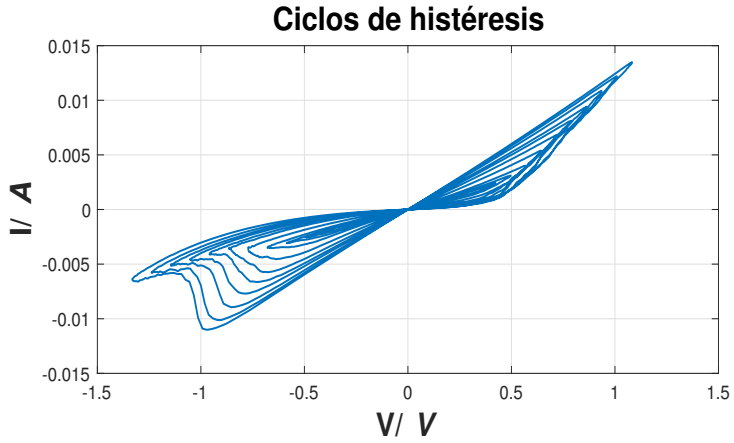

Figura 3.5: Relación Voltaje-Corriente experimental al aplicar la señal II.

3.3. Verificación del modelo

En este apartado, comprobaremos la validez que tiene el modelo de E.Miranda para deducir la corriente que atraviesa al memristor a partir del voltaje experimental, comparando los resultados proporcionados por el modelo con los obtenidos experimentalmente. Este proceso se realizará a través del código del Programa 1 que aparece en el Apéndice B. Sin embargo, para poder utilizar el modelo es necesario determinar previamente los valores de los 8 parámetros del modelo, variables de estado, cuyo valor desconocemos, además de fijar la condición inicial del estado interno del memristor λ_0, que la consideraremos en los dos experimentos igual a cero (lo que se denomina memristor en estado HRS [9]). Es en este momento donde aplicaremos el método de Levenberg-Marquardt estudiado en la Sección 2.4.3.

Siguiendo la misma notación que entonces, nuestra función objetivo ahora es

$$f(x) = \|r(x)\|_2^2 = \sum_{i=1}^{m} r_i^2(x), \tag{3.9}$$

con

$$r_i(x) = I_{exp}(i) - I_{modelo}(i), \tag{3.10}$$

donde $I_{exp}(i)$ es el i-ésimo valor de la corriente recopilado durante un experimento e $I_{modelo}(i)$ es la componente i-ésima resultante al ejecutar la función de Matlab `Intensidad(np,nm,vp,vm,a,R,imax,imin,V)` que aparece en el Programa 1. Las dimensiones de este problema son bastante grandes habiendo recopilado en cada experimento 2400 muestras, de las que se utilizarán 2396 pues las primeras son poco precisas. Luego $m = 2396$ mientrás que $n = 8$, pues el modelo de Miranda posee 8 parámetros desconocidos que hay que determinar.

Como se ha dicho anteriormente, en este caso se ha elegido el método de Levenberg-Marquardt. Esto es debido a la complejidad del modelo, donde métodos del tipo Newton que introdujeran un mayor número de cálculos harían que el proceso fuera demasiado costoso, además de la gran ventaja de poder aplicarlo en Matlab sin necesidad de programar el algoritmo, pues ya se encuentra entre las herramientas que ofrece este programa. De este modo, hemos sido capaces de determinar los parámetros que minimizan las discrepancias entre los resultados del modelo y los resultados experimentales dentro de las tolerancias que hemos impuesto. El código en el que se aplica este procedimiento es el Programa 2 que aparece en el Apéndice B. En él, es necesario introducir unos valores iniciales para empezar la búsqueda de los 8 parámetros. Cuanto más próximos sean estos valores iniciales a los buscados, más rápido y preciso será el resultado obtenido. Los resultados encontrados tras ejecutar varias veces el Programa 2, mejorando los parámetros iniciales introducidos, se han recogido en la siguiente tabla.

	$\eta+/V^{-1}$	$\eta-/V^{-1}$	$V+/V$	$V-/V$
Experimento I	11,92156687	20,28290385	0,81196652	-0,63099599
Experimento II	8,05603987	10,05393047	0,99697151	-0,42929402

	α/V^{-1}	R/Ω	$Imax/A$	$Imin/A$
Experimento I	4,07713109	74,76791879	0,01486760	0,00020335
Experimento II	5,91649173	74,50115922	0,02166648	0,00004179

Tabla 3.1: Los mejores valores determinados, mediante el Programa 2, para las variables de estado del modelo de E. Miranda.

En las siguientes gráficas, se observan visualmente los resultados conseguidos.

✣ Primer experimento

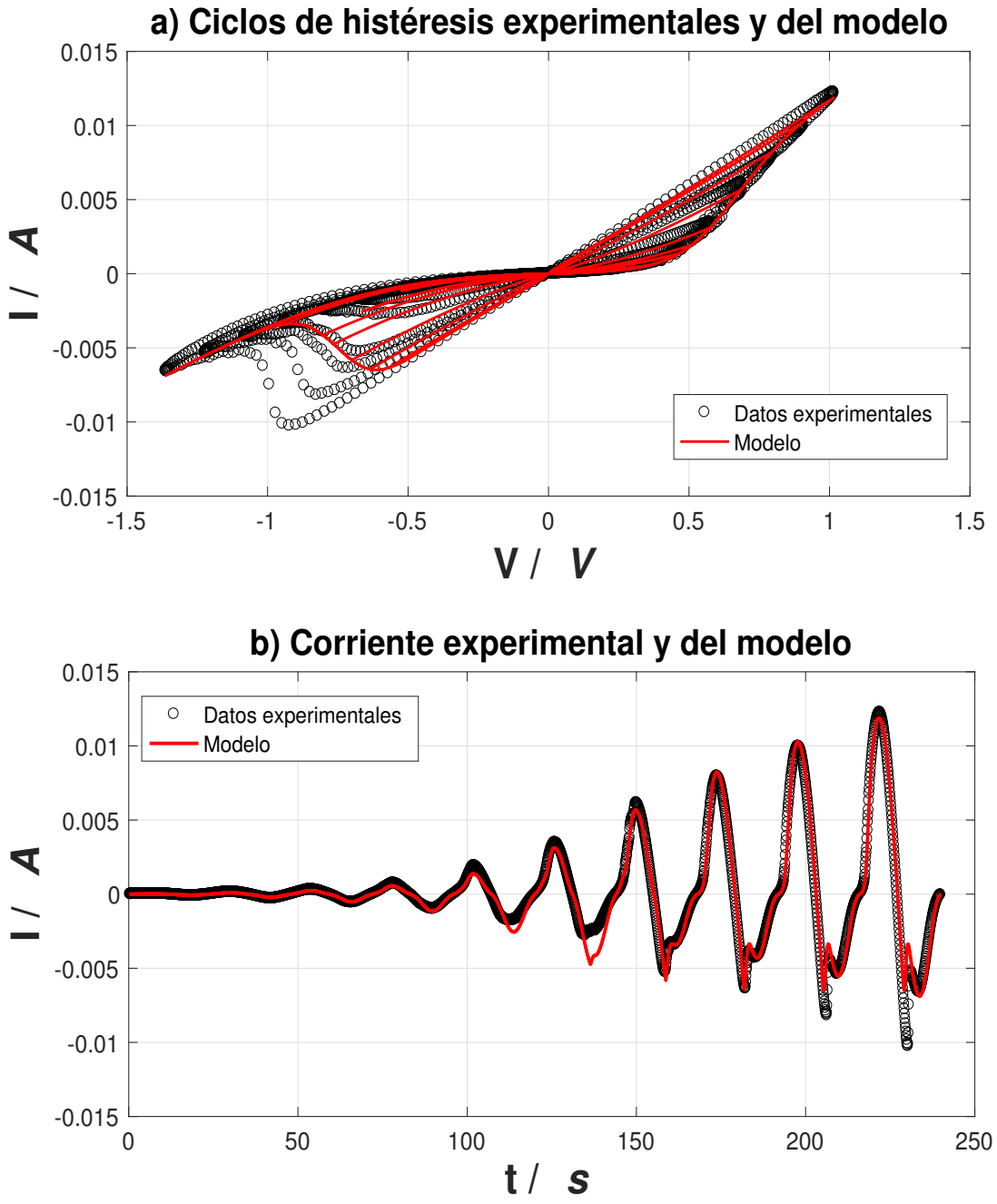

Figura 3.6: Comparativa de los datos recopilados en el primer experimento con los obtenidos mediante el modelo de E. Miranda utilizando los parámetros de la Tabla 3.1 correspondientes a dicho experimento.

✣ Segundo experimento

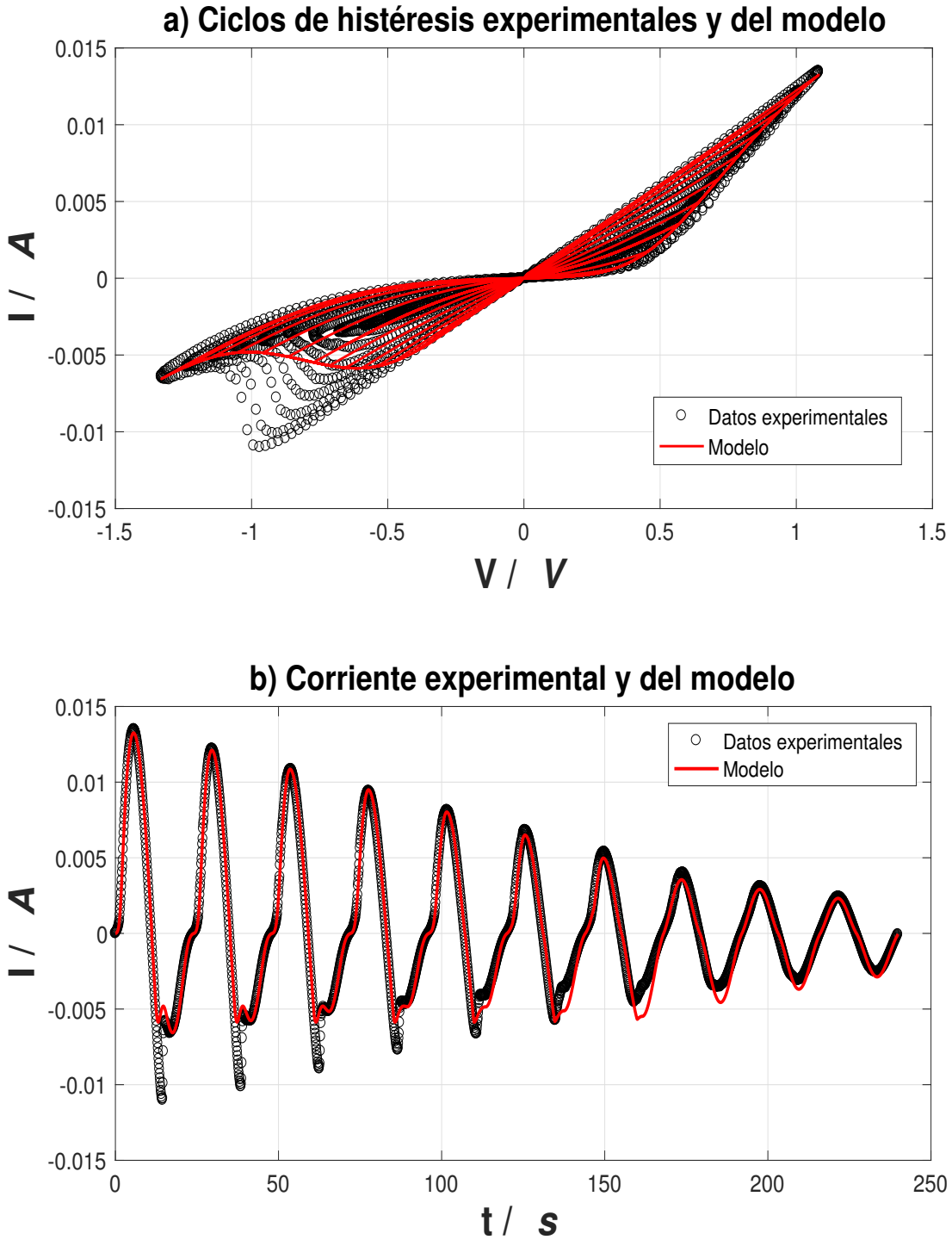

Figura 3.7: Comparativa de los datos recopilados en el segundo experimento con los obtenidos mediante el modelo de E. Miranda utilizando los parámetros de la Tabla 3.1 correspondientes a dicho experimento.

Los errores obtenidos utilizando los parámetros de la Tabla 3.1 son:

	$\varepsilon_{absoluto}$ / A	$\varepsilon_{relativo}$
Experimento I	0,00082834	0,03097019
Experimento II	0,00127891	0,02463943

- En el experimento I:

$$\varepsilon_{absoluto} \approx 8 \cdot 10^{-4} \text{ A}$$
$$\varepsilon_{relativo} \approx 3\,\%$$

- En el experimento II:

$$\varepsilon_{absoluto} \approx 1{,}3 \cdot 10^{-3} \text{ A}$$
$$\varepsilon_{relativo} \approx 2\,\%$$

Estos errores han sido calculados, como se hace habitualmente en los problemas de mínimos cuadrados, empleando la norma euclídea

$$\varepsilon_{absoluto} = \|r(x)\|_2^2 = \sum_{i=1}^{m} [I_{exp}(i) - I_{modelo}(i)]^2, \tag{3.11}$$

$$\varepsilon_{relativo} = \frac{\varepsilon_{absoluto}}{\|I_{exp}\|_2^2} = \frac{\sum_{i=1}^{m} [I_{exp}(i) - I_{modelo}(i)]^2}{\|I_{exp}\|_2^2}, \tag{3.12}$$

donde $I_{modelo}(i) =$ `Intensidad(np,nm,vp,vm,a,R,imax,imin,`V_i`)` y $r(x)$ es el vector residual del problema.

Cabe señalar que los valores de los parámetros recogidos en la Tabla 3.1 podrían ser mejorados ligeramente, ya que en el Programa 2 se han impuesto una serie de criterios de tolerancia, número máximo de iteraciones o diferencia de paso entre iteraciones consecutivas, que determinan cuando acaba el bucle de búsqueda y se podrían utilizar otras combinaciones.

3.4. Conclusiones

En base a los resultados obtenidos, podemos afirmar que el modelo de Miranda aunque consigue construir la estructura de lazos de histéresis típica de los memristores, lo que se denominan lazos pellizcados [2], tiene bastantes limitaciones a la hora de conseguir las corrientes negativas experimentalmente observadas en los lazos de mayor área. Como se muestra en los resultados experimentales, existe una cierta asimetría entre el proceso de formación de canales (set process), que ocurre a corrientes positivas, y el proceso de destrucción de canales (reset process), que ocurre a corrientes negativas [5]. En el modelo de Miranda no se incluye esta asimetría lo que provoca que aparezcan serias discrepancias con lo experimental, especialmente en la zona de reset cuando analizamos todos los lazos a la vez. Sin embargo, si analizáramos de forma individual cada uno de los lazos conseguiríamos un ajuste mucho más preciso, logrando alcanzar incluso las intensidades más negativas, como se puede observar en el caso del lazo de mayor amplitud en la Figura 3.8.

Cabe señalar que desde un punto de vista físico, al estar utilizando el mismo memristor en los dos experimentos, cabría esperar que los parámetros α y R al ajustarlos fueran iguales. Esto no se consigue de forma directa, como vimos en la Tabla 3.1, por lo tanto será necesario hacer una pequeña modificiación del Programa 1 que se basa en imponer valores fijos para dichos parámetros, iguales al valor promedio de los obtenidos en cada experimento (Tabla 3.1), y resolver de nuevo los dos problemas de mínimos cuadrados para determinar el valor de las restantes variables de estado. Los parámetros resultantes se recogen en la Tabla 3.2. Observamos que los valores obtenidos no difieren mucho de los presentados en la Tabla 3.1.

	$\eta+/V^{-1}$	$\eta-/V^{-1}$	$V+/V$	$V-/V$
Experimento I	11,22256015	19,57182728	0,84146960	-0,61213747
Experimento II	8,05358511	10,05245246	0,98199081	-0,44134991

	α/V^{-1}	R/Ω	$Imax/A$	$Imin/A$
Experimento I	4,99681141	74,63453900	0,01291762	0,00009815
Experimento II	4,99681141	74,63453900	0,02605930	0,00009614

Tabla 3.2: Los mejores valores determinados, mediante el Programa 2, para las variables de estado del modelo de E. Miranda considerando que se está usando el mismo memristor, es decir, que α y R deben ser iguales en los dos experimentos.

Los errores con estos parámetros, respecto a lo experimental, son

	$\varepsilon_{absoluto}$ / A	$\varepsilon_{relativo}$
Experimento I	0,00083757	0,03131555
Experimento II	0,00128920	0,02483780

- En el experimento I:
$$\varepsilon_{absoluto} \approx 8 \cdot 10^{-4} \text{ A}$$
$$\varepsilon_{relativo} \approx 3\,\%$$
- En el experimento II:
$$\varepsilon_{absoluto} \approx 1{,}3 \cdot 10^{-3} \text{ A}$$
$$\varepsilon_{relativo} \approx 2\,\%$$

Aunque estos parámetros posean unos errores algo mayores que los obtenidos con los de la Tabla 3.1, los consideraremos más correctos al tener un mayor sentido físico dentro del modelo.

Ahora si realizamos el análisis individual de dos lazos de distinta área tendremos un claro ejemplo de cómo influye la utilización de distintos valores de los parámetros η^- y V^-, pues son los verdaderamente relevantes cómo se explica en [15], en el resultado del modelo, visualizándose los resultados en las gráficas de la Figura 3.8. Para realizar estas gráficas, hemos utilizado los datos recopilados en el primer experimento y calculado los parámetros η^- y V^- que minimizan las discrepancias resultantes desde 215.7s a 239.6 s, valores que corresponden al lazo de mayor amplitud (lazo 9), y desde 119.7s a 143.7 s, valores correspondientes al quinto lazo de mayor área (lazo 5), dejando el resto de parámetros fijos con los valores obtenidos en la Tabla 3.2. Estos parámetros, junto a los obtenidos al ajustar todos los lazos a la vez, se han recogido en la siguiente tabla.

	η-/V^{-1}	V-/V	$\varepsilon_{absoluto}$ / A	$\varepsilon_{relativo}$
Lazo 5	12,63217163	-0,32010122	0,00001327	0,01366019
Lazo 9	37,84753317	-0,92908406	0,00002797	0,00256334
Todos los lazos	19,57182728	-0,61213747	0,00083757	0,03131555

Tabla 3.3: Parámetros encontrados que mejor ajustan, usando el Programa 2, a todos los datos recopilados en el primer experimento así como los que mejor ajustan al lazo mayor (lazo 9) y al quinto lazo mayor (lazo 5). En todos los ajustes sólo se ha permitido que varíen los valores de η^- y V^-. El resto de parámetros tiene los valores: $\eta^+ = 11,22256015\ V^{-1}$, $V^+ = 0,84146960\ V$, $\alpha = 4,99681141\ V^{-1}$, $R = 74,63453900\ \Omega$, $I_{max} = 0,01291762\ A$, $I_{min} = 9,815 \cdot 10^{-5} A$.

Las representaciones gráficas de la variación del ajuste del modelo sobre los datos correspondientes al lazo de mayor amplitud (lazo 9) dependiendo de los parámetros elegidos se muestran a continuación. En ellas, se han utilizado los parámetros de las dos últimas filas de la Tabla 3.3.

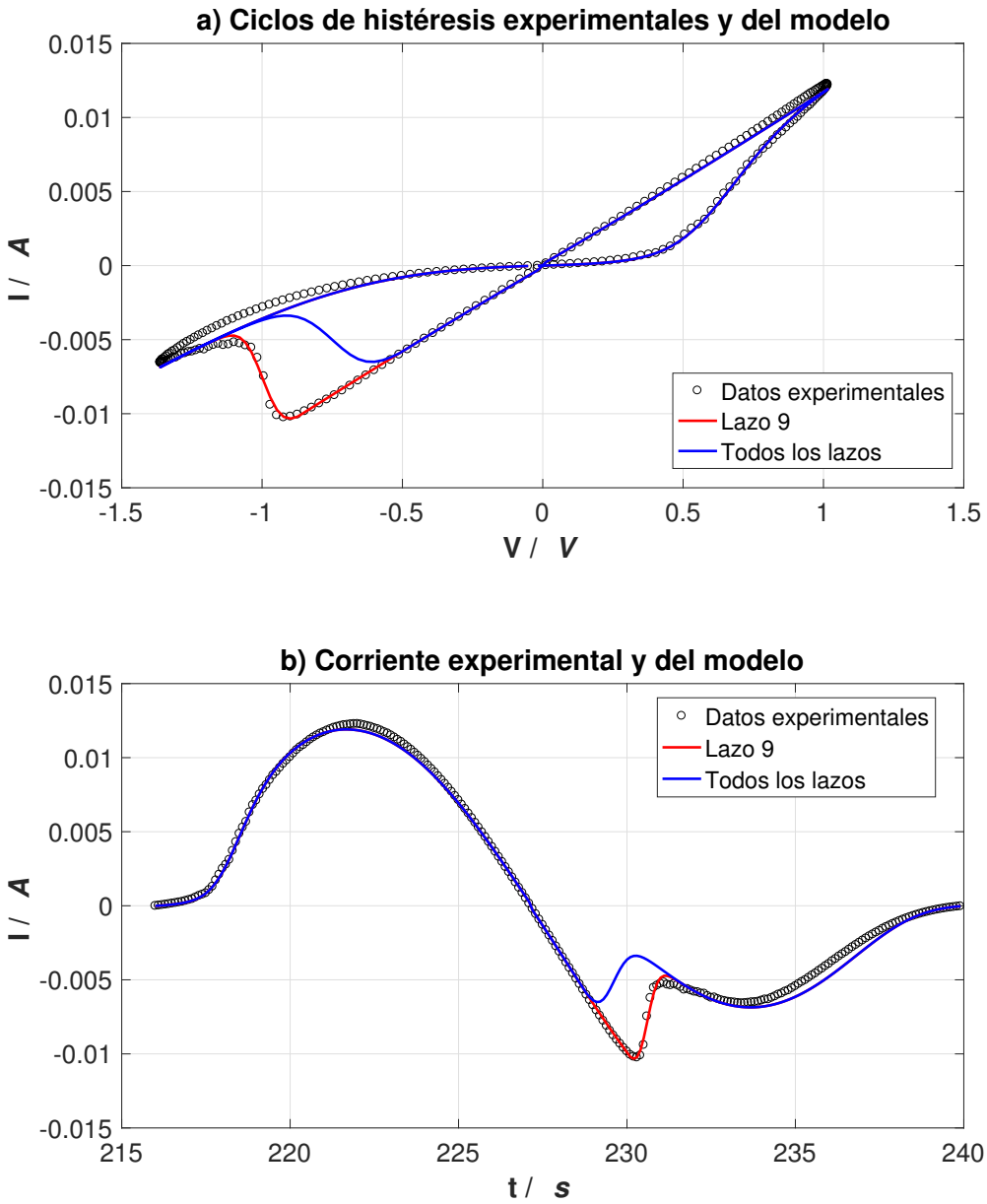

Figura 3.8: Representación de la variación de los resultados obtenidos para los tiempos correspondientes al lazo de mayor amplitud, utilizando los distintos parámetros correspondientes a las dos últimas filas de la Tabla 3.3 en el modelo de Miranda, tanto en la construcción de los lazos de histéresis (a) como en la variación de la corriente eléctrica (b).

Como se puede observar, las discrepancias entre el modelo y lo obtenido experimentalmente en el tiempo correspondiente al lazo de mayor amplitud son mucho menores si consideramos los parámetros que mejor ajustan a dicho lazo, parámetros lazo 9, como cabría esperar. De esta manera, analizando más detenidamente como varían los parámetros η^- y V^- según el lazo se podría encontrar una cierta tendencia, presumiblemente lineal con el tiempo, dando pie a una posible mejora del modelo. Este análisis detallado se ha realizado en el TFG de Física [15] consiguiendo la mejora sugerida.

En resumen, podemos afirmar que el modelo de E. Miranda, aunque de forma general no consigue describir adecuadamente el proceso de reset al aplicar un voltaje sinusoidal creciente o decreciente, mejora considerablemente cuando los parámetros del modelo se ajustan para analizar de forma individual cada lazo de histéresis, por lo que se propone un camino de investigación para mejorar dicho modelo.

Apéndice A:
La pseudoinversa de una matriz

La pseudoinversa de una matriz, también llamada inversa de Moore-Penrose, es muy útil a la hora de resolver el problema lineal de mínimos cuadrados. Fue descubierta por Bjerhammar y Penrose [13], quienes la caracterizaron siguiendo el siguiente teorema.

Teorema A.1. ***Condiciones de Penrose***
Sea $A \in \mathbb{R}^{m\times n}$. La pseudoinversa $A^+ \in \mathbb{R}^{n\times m}$ de la matriz A es la única matriz que satisface las siguientes condiciones:

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $A^+A = (A^+A)^T$
4. $AA^+ = (AA^+)^T$

Las propiedades más relevantes que cumplen estas matrices se recogen en el siguiente teorema.

Teorema A.2. *Sea $A \in \mathbb{R}^{m\times n}$ y A^+ su matriz pseudoinversa. Se verifica*

1. $(A^+)^+ = A$;
2. $(A^+)^T = (A^T)^+$;
3. $(\alpha A)^+ = \alpha^+ A^+$;
4. $(A^TA)^+ = A^+(A^+)^T$
5. *Si U y V son matrices ortogonales entonces* $(UAV^T)^+ = VA^+U^T$
6. *Si $A = \sum_i A_i$, donde $A_iA_j^T = 0, A_i^TA_j = 0$ para $i \neq j$ entonces $A^+ = \sum_i A_i^+$.*
7. *Si A es normal, es decir, $AA^T = A^TA$, entonces $A^+A = A^+$.*
8. *Las matrices A, A^T, A^+ y A^+A tienen el mismo rango, que coincide con la traza de A^+A.*

La demostración de estas propiedades se pueden encontrar en [13].

Observación A.3. Distinguiendo casos concretos podemos enunciar algunas propiedades adicionales de la matriz pseudoinversa.

- Si $m = n$ y A es no singular entonces se cumple $A^+ = A^{-1}$.
- Si $m > n$ y las columnas de A son linealmente independientes, lo que se conoce como rango columna completo, entonces $A^+ = (A^T A)^{-1} A^T$.
- Si $m < n$ y las filas de A son linealmente independientes, lo que se conoce como rango fila completo, entonces $A^+ = A^T (AA^T)^{-1}$.
- Si A no tiene rango completo entonces se utiliza la descomposición en valores singulares de A. Si esta es $A = U\Sigma V^T$, entonces $A^+ = V\Sigma^+ U^T$.

Es conveniente recordar las siguientes propiedades en relación con la descomposición en valores singulares de una matriz

- ☞ Los valores singulares de A son las raíces cuadradas positivas de los valores propios no negativos de $A^T A$. Se denotan por σ_i con $i = 1, \ldots, r = rang(A)$.
- ☞ La matriz $\Sigma \in \mathbb{R}^{m \times n}$ es una matriz diagonal en el sentido amplio, es decir,

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \sigma_n \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{matrix} \uparrow \\ \\ n \\ \\ \downarrow \\ \uparrow \\ m-n \\ \downarrow \end{matrix} \tag{A.1}$$

  con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_n = 0$.
- ☞ La matriz V es una matriz ortogonal cuyas columnas v_i son los vectores propios normalizados de $A^T A$ en el orden dado por los valores singulares σ_i.
- ☞ La matriz U es una matriz ortogonal cuyas columnas u_i son los vectores propios normalizados de AA^T en el orden dado por los valores singulares. Las r primeras columnas se pueden calcular como $u_i = \frac{1}{\sigma_i} A v_i$ y las siguientes $m - r$ se eligen de manera que sean ortonormales (usando Gram-Schmidt si es necesario).

Por lo tanto, a la hora de resolver el problema lineal de mínimos cuadrados donde las ecuaciones normales son

$$A^T A x_0 = A^T b, \tag{A.2}$$

distinguiremos

- Si A^TA es invertible entonces $x_0 = (A^TA)^{-1}A^Tb = A^+b$.
- Si A^TA no es invertible entonces el problema de mínimos cuadrados tiene infinitas soluciones. En este caso, se puede demostrar que, usando la SVD, $x_0 = A^+b = V\Sigma^+U^Tb$ proporciona una solución de las ecuaciones normales con la propiedad de tener la norma euclídea mínima.

Apéndice B: Código Matlab

B.1. Cálculo de la corriente según el modelo de E. Miranda

Mediante el primer programa de Matlab seremos capaces de calcular la corriente que atravesaría al memristor a partir de los voltajes experimentales, según el modelo de Miranda. Los resultados obtenidos con este programa se han utilizado tanto para mostrar numérica y gráficamente como para analizar las discrepancias entre los datos experimentales y los conseguidos con el modelo.

Programa 1: Cálculo de la intensidad de corriente

```
1  %Programa 1: Cálculo de la intensidad según el modelo de ...
       E.Miranda.
2  %Es necesario introducir np,nm,vp,vm,a,R,imax,imin
3  function [Intensidad]=I(y)
4      datos %tabla que almacena los datos experimentales del ...
           primer experimento.
5      %Variables de estado:
6      np=y(1); %Velocidad de creación
7      nm=y(2); %Velocidad de destrucción
8      vp=y(3); %Voltaje medio de creación
9      vm=y(4); %Voltaje medio de destrucción
10     a=y(5);  %alpha (mecanismo de conducción del memristor)
11     R=y(6);  %Resistencia
12     imax=y(7); %Corriente máxima que atraviesa al memristor
13     imin=y(8); %Corriente mínima que atraviesa al memristor
14     V=x(:,8); %Voltajes experimentales
15     %Condición inicial
16     L0=0; %Inicialmente el memristor está en estado HRS.
17
18 %Paso 1: Creamos las curvas de formación y destrucción de los ...
       filamentos conductores (CFs)
19     %gp --> Gamma+
20     %gm --> Gamma-
21     [gp,gm]=gamma(V,np,nm,vp,vm);
```

```
%Paso 2: Calculamos la variable de estado interno del ...
   memristor L(t)--> lambda(t)
    L=zeros(length(V),1); %inicializamos el vector
    for i=1:length(V)
        L(i)=min(gm(i),max(L0,gp(i)));
        L0=L(i);
    end

%Paso 3: Calculamos la amplitud de corriente del diodo según ...
   el modelo Io
   Io=ampli(L,imax,imin);
%Paso 4: Calculamos la corriente que atraviesa al memristor ...
   usando la aproximación de Hermite-Padé de la función de ...
   Lambert.
Intensidad=sign(V).*((1/(a*R)) ...
           .*w(a.*R.*Io.*exp(a.*(abs(V)+R.*Io)))-Io);

%Representación gráfica de los resultados obtenidos

figure(1)%Lambda vs Voltaje
clf
plot(V,L)
xlabel('V')
ylabel('Lambda')
hold on
plot(V,gp)
plot(V,gm)

figure(2)%Corriente vs Voltaje
clf
plot(V,Intensidad,'red') %En rojo con las corrientes del modelo
hold on
plot(x(:,8),x(:,5),'blue') %En azul con las corrientes ...
   experimentales
xlabel('V')
ylabel('I')
legend('Modelo','Experimental')

figure(3) %Corriente vs tiempo
clf
plot(x(:,7),Intensidad,'red') %En rojo con las corrientes del ...
   modelo
hold on
plot(x(:,7),x(:,5),'blue')%En azul con las corrientes ...
   experimentales
xlabel('t')
ylabel('I')
legend('Modelo','Experimental')

figure(5) %ln(|Corriente|) vs Voltaje
clf
semilogy(x(:,8),abs(Intensidad),'red')%En ROJO LA DEL MODELO
```

```
hold on
semilogy(x(:,8),abs(x(:,5)),'blue') % en AZUL LA EXPERIMENTAL
xlabel('V')
ylabel('ln(abs(I))')
legend('Modelo','Experimental')
end

%Funciones auxiliares necesarias en la ejecución del modelo:

%Función Gamma
function [gp,gm]=gamma(V,np,nm,vp,vm)
gp=1./(1+exp(-np.*(V-vp)));
gm=1./(1+exp(-nm.*(V-vm)));
end
%Función Lambert W
function lambert=w(j)
lambert = log(1+j).*(1-(log(1+log(1+j))./(2+log(1+j))));
end
%I0(lambda)
function Io=ampli(L,imax,imin)
 Io = imax.*L+imin.*(1-L);
end
```

B.2. Ajuste de los parámetros del modelo de E.Miranda

En el segundo programa de Matlab determinamos los parámetros que minimizan lo máximo posible las discrepancias de las corrientes conseguidas mediante el modelo de Miranda y las recogidas experimentalmente. Para ello, hacemos uso del método de mínimos cuadrados de Levenberg-Marquardt, que está implementado en la función `lsqnonlin` de Matlab. Una vez encontrados los valores de los parámetros, hemos procedido a visualizar los resultados a través de diferentes gráficas.

Programa 2: Ajuste de los parámetros del modelo

```
%Programa 2: Búsqueda de los parametros del modelo de E. ...
   Miranda con los que minimizar los errores con los datos ...
   experimentales.
datos; %tabla que almacena los datos experimentales del primer ...
   experimento.
%Valores inicial (p) y final (f) que tomamos para realizar el ...
   ajuste (estos valores se cambiarán para poder hacer el ...
   ajuste de solo uno de los lazos).
p=1;
q=2396;
V=x(p:q,8); %Voltajes experimentales
```

```
7  Iexp=x(p:q,5); %Corrientes experimentales
8  %Los parametros desconocidos del modelo se recogen en el ...
      vector y en el siguiente orden: ...
      y0=[np0,nm0,vp0,vm0,a0,R0,imax0,imin0]
9  %Hay que dar un valor inicial a estos parametros los cuales ...
      deben ser razonables para conseguir que el método de ajuste ...
      sea bueno (que alcance el mínimo global del problema y no ...
      uno local).
10 %En el caso del primer experimento se han elegido:
11 y0=[12,20,0.81,-0.63,5,74.6,0.0148,0.0002];
12 [IntensidadMemristor,Io]=Intensidad1(y0); %Se calcula la ...
      corriente que atraviesa al memristor según el modelo.
13 %Para ello se utiliza el programa Intensidad1, que es idéntico ...
      al Programa 1 sin las gráficas.
14
15 %Calculamos el error inicial del modelo con respecto a los ...
      experimentos usando la norma al cuadrado (lo habitual en el ...
      problema de mínimos cuadrados):
16 r=x(p:q,5)-IntensidadMemristor;
17 f=(norm(r)^2)
18
19 %Usamos el método de Levenberg-Marquardt para determinar los ...
      parámetros que mejor ajusten al modelo (haciendo el error ...
      lo más pequeño posible).
20 fun=@(y)Intensidad1(y)-Iexp;
21 options.Algorithm='levenberg-marquardt';
22 options = optimoptions('lsqnonlin','Display','final');
23 %Aumentamos los valores de la tolerancia preestablecida en el ...
      algoritmo para encontrar una mejor precisión de los parámetros.
24 options.OptimalityTolerance=1e-35;
25 options.FunctionTolerance=1e-25;
26 options.StepTolerance=1e-25;
27 options.MaxFunctionEvaluations=500;
28 [z,resnorm]=lsqnonlin(fun,y0,[],[],options);
29 LevenbergCrec=[z,resnorm]; %Almacenamos la solución en este ...
      vector, tanto los parámetros como el mínimo error alcanzado ...
      entre el modelo y lo experimental.
30 IntFinal2=Intensidad1(z); %Corriente del modelo con los ...
      parámetro ajustados
31
32 %Gráficas:
33
34 figure(1) %(Corriente experimental vs Voltaje)+(Corriente ...
      modelo vs Voltaje)
35 clf
36 %Mediante círculos negros marcamos los resultados experimentales.
37 for i = 1:length(V) %Bucle con el que observar como se va ...
      pintando la gráfica.
38     plot(V(1:i),Iexp(1:i),'ko')
39     pause(0.000000001)
40 end
41 hold on
```

```
42 %Mediante una línea roja marcamos los resultados del modelo ...
       (Levenberg)
43 for i = 1:length(V)
44     plot(V(1:i),IntFinal2(1:i),'r-')
45     pause(0.0001)
46 end
47 xlabel('V')
48 ylabel('I')
49 title('Ciclos de histéresis experimentales y del modelo')
50 legend('Datos experimentales', 'Modelo')
51 hold off
52
53 figure(2) %Corriente vs tiempo
54 clf
55 plot(x(p:q,7),Iexp,'ko') %En círculos lo experimental
56 hold on
57 plot(x(p:q,7),IntFinal2,'r-') % en rojo el modelo
58 xlabel('t')
59 ylabel('I')
60 title('Corriente experimental y del modelo')
61 legend('Datos experimentales', 'Modelo')
62 hold off
```

Bibliografía

[1] Björck, A., *Numerical Methods for least squares problems*, SIAM, Philadelphia, 1996.

[2] Chua, L., Sirakoulis G.C. y Adamatzky, A., *Handbook of Memristor Networks*, Springer, Switzerland, pp.165-178, 2018.

[3] Dennis, J.E., Gay D.M. y Welsch, R.E., *An Adaptive Nonlinear Least-Squares Algorithm*, ACM Trans. Math. Software, Vol.7, No 3, pp.348-368, 1981.

[4] Dennis, J.E. y Schnabel, R.B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996. Corrección de la primera publicación en Prentice Hall, Englewood Cliffs, NJ, 1983.

[5] Dueñas, S., Castán, H., Ossorio, O.G. y García, H., *Dynamics of set and reset processess on resistive switching memories*, Microelectronic Engineering, Universidad de Valladolid, España, 2019.

[6] Gill, P.E., Murray, W. y Wright, M.H., *Practical Optimization*, Academic Press, pp.100-102, 1981.

[7] Golub, G. y Van Loan, C., *Matrix computations*, 3rd edition, Johns Hopkins University Press, Baltimore, 1996.

[8] Lindström, P. y Wedin, P.A., *A new line search algorithm for unconstrained nonlinear least squares problems*, Math. Programming, pp.268-296, 1984.

[9] Miranda, E., *Compact Model for the Major and Minor Hysteretic I–V Loops in Nonlinear Memristive Devices*, IEEE Trans. Nanotechnol., Vol. 14, No. 5, pp.787–789, 2015.

[10] Moré, J.J., *The Levenberg-Marquardt algorithm: Implementation and theory*, Conference on Numerical Analysis, Dundee, 1977.

[11] Nocedal, J. y Wright, S.J., *Numerical Optimization*, 2nd edition, Springer, New York, 2006.

[12] Ortega, J.M. y Rheinboldt, W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, Philadelphia, pp.257 y 491, 2000.

[13] Penrose, R., *A generalized inverse for matrices*, Proc. Cambridge Phil. Soc. 51, pp.406-413, 1955.

[14] Ruhe, A., *Accelerated Gauss-Newton algorithms for nonlinear least squares problems*, BIT 19, pp.356-367, 1979.

[15] Santa Cruz González, C., *Modelización matemática de curvas experimentales en regímenes estacionarios y de pequeña señal de memristores*, Trabajo de Fin de Grado, Universidad de Valladolid, 2020.

[16] Watkins, D.S., *Fundamentals of Matrix Computations*, 2nd edition, John Wiley & Sons, New York, 2002.